SMART INNOVATION, SYSTEMS AND TECHNOLOGIES 17

Alejandro Peña-Ayala
Editor

Intelligent and Adaptive Educational-Learning Systems

Achievements and Trends

Springer

Smart Innovation, Systems and Technologies 17

Editors-in-Chief

Prof. Robert J. Howlett
KES International
PO Box 2115
Shoreham-by-sea
BN43 9AF
UK
E-mail: rjhowlett@kesinternational.org

Prof. Lakhmi C. Jain
School of Electrical and Information Engineering
University of South Australia
Adelaide
South Australia SA 5095
Australia
E-mail: Lakhmi.jain@unisa.edu.au

For further volumes:
http://www.springer.com/series/8767

Alejandro Peña-Ayala (Ed.)

Intelligent and Adaptive Educational-Learning Systems

Achievements and Trends

Springer

Editor
Alejandro Peña-Ayala
WOLNM
Leyes Reforma
Mexico

ISSN 2190-3018 e-ISSN 2190-3026
ISBN 978-3-642-30170-4 e-ISBN 978-3-642-30171-1
DOI 10.1007/978-3-642-30171-1
Springer Heidelberg New York Dordrecht London

Library of Congress Control Number: 2012937668

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Preface

Educational learning systems (ELS) represent computer-based approaches devoted to spread educational services for teaching and learning mainly through the Internet. When the development of ELS takes into account artificial intelligence techniques (e.g., acquiring and representing knowledge, make inferences and automatic learning) they become intelligent. ELS are adaptive, once they pursue to adapt themselves to satisfy users' needs, such as: navigation, interaction, content authoring and delivering, sequencing, assessment, evaluation, assistance, supervision and collaboration. Hence, ELS that include some kind of intelligent and adaptive functionality are called: intelligent and adaptive ELS (IALES).

This book reveals a sample of current work in the IALES, where researchers and practitioners of fields such as pedagogy, education, computer sciences, artificial intelligence, and graphic design join efforts to outcome frameworks, models, methods, systems and approaches for innovate the provision of education and enhance the learning of students. According to the nature of the contributions accepted for this volume, four kinds of topics are presented as follows:

- Modeling: An essential component of any IAELS is the user model. It depicts relevant cognitive and personality traits of the student, the assessment of her/his performance, the acquired domain knowledge and other useful attributes in order the IALES to behave adaptive to tailor user's learning needs.
- Content: Content represents the raw material and the main source of stimuli for students in order they to acquire knowledge, develop skills and gain experiences to accomplish some level of competence in a given educational domain.
- Virtuallity: Modern user-system interfaces and technologies engage students to work in virtual environments that catch their senses and challenge their cognitive faculties in such a way they represent a new educational paradigm.
- Applications: Several sorts of approaches compose the scope of IAELS such as: metacognition, educational system architectures, collaborative learning, educational data mining and case studies.

This volume is the result of one year of effort, where more than forty chapters were rigorous peer reviewed by a set of ninety reviewers. After several cycles of chapter submission, revision and tuning based on the KES International quality principles, twenty works were approved, edited as chapters and organized according to the prior topics. So the first part corresponds to *modeling* and includes chapters 1 to 5; the second part represents *content* and embraces chapters 6 to 10; the third part concerns to *virtuallity* and holds chapters 11 to 14; the fourth part is related to *applications* and contains chapters 15 to 20. A profile of the chapters is given next:

1. Chapter one introduces an affective behavior model to point out a student's affect state by means of a dynamic Bayesian network and a cognitive model of emotions.
2. Chapter two presents an adaptive learning environment model composed by four models (e.g., domain, learner, course structuring, adaptation) in order to set adaptive learning curriculum.
3. Chapter three proposes a proactive sequencing based on a fuzzy-causal student model to estimate learning outcomes that different content about a given concept of the domain knowledge produce on the apprenticeship of the student for choosing the most profitable option.
4. Chapter four aims at applying mining process to learner models for finding out rules from event logs. The approach combines learning styles with process mining procedures.
5. Chapter five aims at using a learning style index to find out effective ways to learn. Moreover, the work advices tutor to adopt suitable content for efficient teaching.
6. Chapter six reports the experience gained with the use of the GRAPPLE, an environment that holds a common user model framework, where structured content is authored and adaptation is set as guidance and personalized material.
7. Chapter seven aims at adaptive content selection by means of an adaptation model, which uses a decision-based approach to adaptively choose learning objects in educational hypermedia systems.
8. Chapter eight outlines a collaborative adaptive learning tool, which is able to produce several instances of a learning object by the parameterization of some features through metadata.
9. Chapter nine shares a case study about the use of an adaptive learning management system and authoring tool to support the design of adaptive and reusable courses.
10. Chapter ten pursues the reuse of intelligent tutoring systems; thereby it implements them as if they were learning objects by means of the Sharable Content Object Reference Model.
11. Chapter eleven details how three-dimensional virtual worlds are suitable environments to be collaboratively used by a group of peers aimed to accomplish a common goal, such as writing.
12. Chapter twelve engages students to develop skills and gain knowledge within a smart home domain, which is intended to anticipate and meet inhabitant's needs as they adapt to changing goals and preferences.

13. Chapter thirteen relates a naval training experience, where conscripts were trained by means of a closed-loop adaptive training system that delivers tactical air controllers instruction and provides additional practice lectures.
14. Chapter fourteen implements a cognitive tutoring agent that holds episodic, emotional, procedural and causal learning capabilities, which are used during its interactions with users to enhance the support it provides.
15. Chapter fifteen outlines an approach to adapt agent prompts as scaffolding of reflection at two levels, generic and specific, that is implemented to support students' learning-by-teaching activities.
16. The chapter sixteen aims at triggering self-regulation to encourage users of an educational learning system to acquire higher order knowledge by means of using a dynamic modeling environment.
17. Chapter seventeen outlines a seamless Web-mediated training courseware design model that encourages novice courseware authors to deliver their own adaptive educational-learning systems.
18. Chapter eighteen examines whether the provision of illusionary sense of control, implicit in collaborative learning, is perceived as current control and cause intrinsic motivation towards better work.
19. Chapter nineteen points out an intelligent system for modeling and supporting academic educational processes, which aims at evaluating and refining university curricula in terms of best possible accumulative grade point average.
20. Chapter twenty evaluates three areas of the e-learning process (e.g., technological, business, educational) and presents a case study about how motivation is a key component to encourage students to get complete e-learning courses.

I wish to express my great attitude to all authors, all reviewers, the Springer editorial team, and the editors Prof. Thomas Ditzinger and Prof. Lakhmi C. Jain for their respective collaboration to accomplish this work.

Moreover, I acknowledge the support provided by the National Council of Science and Technology (CONACYT) and the National Polytechnic Institute (IPN) of Mexico by means of the grants: CONACYT 118862, CONACYT-SNI-36453, CONACYT 118962-162727, IPN-SeAca/COTEPABE/144/11, IPN-COFAA-SIBE, IPN-SIP-20120266, IPN-SIP-EDI: SIP/DI/DOPI/EDI-0505/11.

The last but not least, I appreciate the strength given by my Father, Brother Jesus and Helper, as part of the research projects of World Outreach Light to the Nations Ministries (WOLNM).

March 2012 Alejandro Peña-Ayala

Contents

Part I: Modeling

Part II: Content

Part I
Modeling

Chapter 1
Affective Modeling for an Intelligent Educational Environment

Yasmín Hernández[1], L. Enrique Sucar[2], and Gustavo Arroyo-Figueroa[1]

[1] Instituto de Investigaciones Eléctricas, Reforma 113, Cuernavaca, México
{myhp,garroyo}@iie.org.mx
[2] Instituto Nacional de Astrofísica, Óptica y Electrónica,
Luis Enrique Erro 1, Tonantzintla, México
esucar@inaoep.mx

Abstract. Emotions have a ubiquitous role in education and play a key role in learning and motivation. A motivated student learns in a better way than an indifferent student. There is evidence that tutors look at and react to the emotional state of students to motivate them and improve their learning. As regards computers, they have made a contribution in education. There are programs to teach almost any subject matter, but the real challenge consists in providing personalized support to human learning in view of previous knowledge and affective states to achieve an adaptive and intelligent educational-learning system. We have developed an affective behavior model that considers the affect and the knowledge state to provide students with an adaptive and intelligent instruction. The affective behavior model has been integrated into an environment to learn robotics. The instruction is presented by an animated intelligent agent. The affective behavior model maintains an intelligent representation of the student's affect state to adapt the instruction by means of a dynamic Bayesian network (DBN). The affect diagnosis is based on the Cognitive Model of Emotions (CME) and on the five-factor model of personality. The model was evaluated and the results show a high precision in the affective student model and on students learning. We present the model to endow educational environments with affective behavior wherein students' affect is reflected on the user-system interactions. Our affective student model sets an intelligent representation of the student. We present results from the model evaluations.

1.1 Introduction

Emotions have been recognized as an important component in motivation and learning. There is evidence that experienced human tutors monitor and react to the emotional state of the students in order to motivate them and to improve their learning process (Johnson et al. 2000, Qu et al. 2005).

A. Peña-Ayala (Ed.): Intelligent and Adaptive ELS, SIST 17, pp. 3–24.
springerlink.com

Recently, there has been extensive work on modeling student emotions in intelligent tutoring systems; an example of this kind of research can be found in (Conati and Mclaren 2009). However, there have been only limited attempts to integrate information on student affect into the tutorial decisions (Zacharov et al. 2008, Faivre et al. 2003, Murray and VanLehn 2000).

If we want to consider the student's affective state in the tutorial actions, an important problem is to identify the best tutorial action, given both the students' knowledge and affective state. In this chapter, we describe an approach to tackle this problem. We have developed an affective behavior model (ABM) that considers both the knowledge and the affective state of the student to provide students with adaptive and intelligent instruction. We have designed the ABM based on interviews with qualified teachers with the purpose of understanding the reason teachers carry out their actions according to the state of affect and the knowledge of the students. This work is one of the first attempts to build an affective tutor, in particular, based on an extensive study with teachers. In the literature there are very few studies reported with as many teachers participating (Alexander et al. 2005).

The affective behavior model maintains an affective student model by means of a DBN, and it is used to adapt the instruction. The affect prediction is based on contextual information as proposed by the well-known CME (Ortony et al. 1988). The affective student model also takes into account the theory stated by the Five-Factor model of personality (Costa and McCrae 1992).

Although sometimes emotion and mood are used interchangeably, we are making a distinction between them. We consider *mood* as representing an emotional state with longer duration time, whereas we consider *emotion* as a state with shorter time duration. These two states have impact on one another and interact in several ways. In this work, we include only emotions but we are planning on including mood in our student model. Herein, we will use affective state to mean emotional state. We decided to use *affect* instead of *emotion* as stated by (Picard 2000), who affirms that *emotion* has a negative connotation, whereas *affect* does not.

For testing the affective student model, the ABM is being integrated into an environment to learn mobile robotics (Noguez and Sucar 2005); the results are encouraging since they show a high precision of the affective student model. In this chapter, we present the affective student model.

1.2 Trends and Related Work

In the complex task of endowing computers with affective behavior there are several issues and proposals. Some approaches focus on providing computer programs with moods, temperaments, etc., while other proposals try to understand the users' affective state and react accordingly. All of these proposals try to make an adaptive and convincing user-computer interaction. In the educational field the final aim is endowing educational programs with emotional abilities to help students learn.

In order to understand the students' affect there are several proposals. Some proposals are based on corporal and biological signals, such as skin conductivity, blood volume pressure, muscle tension activity and other proposals are based on facial expression. In the literature, we can find works with the latter approach, for example, in (Abbasi et al. 2007) a relationship between facial expressions and affective states is established. They conducted a study videotaping students and asking them about their affective state.

In (Dragon et al. 2008) a proposal is described which includes technology to collect information about emotional states with real-time and multimodal sensors. They use a pressure mouse to detect the increase of pressure related to the increase of levels of frustration. A posture analysis seat which works with pattern recognition algorithms to identify interest and boredom is also used. In addition, they detect the skin conductivity by means of a sensor in a kind of glove; the detected signal is related to attention-getting events. Finally, a facial expression camera is integrated. This proposal looks to integrate emotion detection within an intelligent tutor as part of learning in a natural classroom setting.

A wearable camera system is presented in (Teeters et al. 2006). The camera analyses, in real-time, the facial expressions and head gestures of its wearer and infers six affective-cognitive states. These are: agreeing, disagreeing, interested, confused, concentrating and thinking.

Conversely, other approaches based their investigations on theoretical models of emotions with strong support in psychology. These models establish emotions given certain circumstances. For example, (Kort et al. 2001) proposes a pedagogical model of emotions. They state: "Emotions and learning are closely related, and that through the learning process, the students cross over several steps related to emotion dimensions". Another theoretical model is the CME (Ortony et al. 1988) which states that emotions emerge as a matching process between goals, principles and preferences with the current situation.

The CME was used in the design of the animated agent PAT, pedagogical and affective tutor (Jaques and Viccari 2005). PAT interacts with students by means of emotional behavior. The agent recognizes students' affective state given their actions and tries to motivate them with facial, corporal and textual communication.

However, this task is extremely difficult, and therefore there are many investigations attempting to explain the relationship between learning and affect. For example, in (D´Mello et al. 2008) a survey is presented comparing students' self-reports and teachers' judgments during several tutoring sessions, trying to establish a relationship between situation and affective states. In (Lehman et al. 2008) the relationship between affective state and tutor's actions is investigated. There is also some research trying to endow the tutor with personality, such as in (Kim et al. 2007) wherein the impact of different agents' personalities on students is analyzed.

Despite the importance of emotions in learning has been stated long time ago, the affective computing and particularly its application to learning environment is recent. In view of the related works, we can see how many issues are immersed in the affect processing and how much has to be done in order to have a model to respond with a suitable action and at the appropriate pedagogical time.

The affect research focuses on relationships between emotion, cognition and learning. The current research involve physical sensors o theoretical models to observe the emotions that occur during learning, investigating relationships between emotions and learning gains, modeling the temporal dynamics of the emotions, identifying cognitive, bodily and linguistic indicators of emotional expressions.

We are interested in developing a comprehensive model to detect students' emotions and to act accordingly; but a difference was reported by (Dragon et al. 2008) and (Teeters et al. 2006), who are interested in physical signs of emotions, our first step is to understand the cognitive basis of emotions and its relationship with learning as reported by (Kort et al. 2001) and (Ortony et al. 1988).

Our work reacts before students' emotions more than to show emotions as is investigated by (Jaques and Viccari 2005) and (Kim et al. 2007). We present our proposal to model affective behavior in learning environments in the next section.

1.3 Modeling Affective Behavior

Traditionally, an intelligent educational system decides what and how to teach based on a representation of the student's knowledge. However, there is evidence that experienced human tutors manage the affective state of students to motivate them and improve their learning process (Johnson et al. 2000, Qu et al. 2005). Thus, the student representation structure needs to be augmented to include knowledge about the affective state. An affective model which makes decisions with base on the students' affective state is also needed. In that way, the students can be provided with a tutorial action which fulfills knowledge requirements, and at the same time is appropriate with the student's affective state. Fig. 1.1 shows the general architecture of an intelligent educational system with affective modeling.

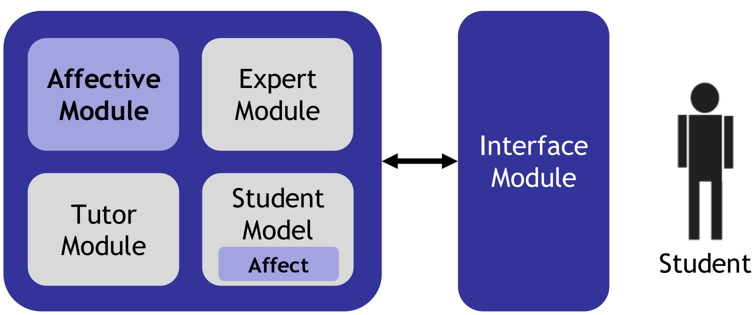


Fig. 1.1 Architecture of an intelligent educational system with affective modeling. The affective model extends the basic architecture of an intelligent educational system as it integrates information about the students' affect and includes an affective module to reason with this affective state, and in this way provides students with an affective and pedagogically suitable response

This architecture includes information about the student's affect in the student model; and it also includes an affective module. This new module contains knowledge to permit reasoning with the student affect. This architecture is based on the one proposed for intelligent tutoring systems (Burns and Capps 1988).

In the context of this work, the process of endowing educational systems with affective behavior includes two aspects: 1) understanding the affective state of the student; 2) deciding the tutorial action to be presented to the student in view of the student's affective and knowledge state. With these two aspects as the goal, we have developed an ABM for intelligent educational systems. The ABM is composed of two main components: the *affective student model* and the *affective tutor model*. A flow diagram of the ABM is presented in Fig. 1.2.

The ABM is set to enable intelligent educational systems to include affective responses in their pedagogical actions. The ABM relies on three elements for selecting the tutorial action to be presented to students: a model of the student's current knowledge (*pedagogical student model* in Fig. 1.2), a model of the student's current affect (*affective student model* in Fig. 1.2), and the tutorial situation.

The tutor module receives these three elements and produces an affective action and a pedagogical action rooted in pedagogical and affective models. The pedagogical action supports the students' learning and the affective action boosts students' morale in the current situation. The two actions are then integrated into the actual tutorial action delivered to the student through the interface module.

The affective action helps the pedagogical model to establish the next pedagogical action, and it also helps the interface module to establish the physical presentation of the pedagogical action. The decision of selecting the affective action first and using it to guide the selection of the pedagogical actions is based on feedback from the teachers in our investigations. Twenty teachers participated in our studies; they stated that they first observe the affect and motivation of students and then subsequently decide on the pedagogical strategies (Hernández et al. 2009, Hernández et al. 2009b).

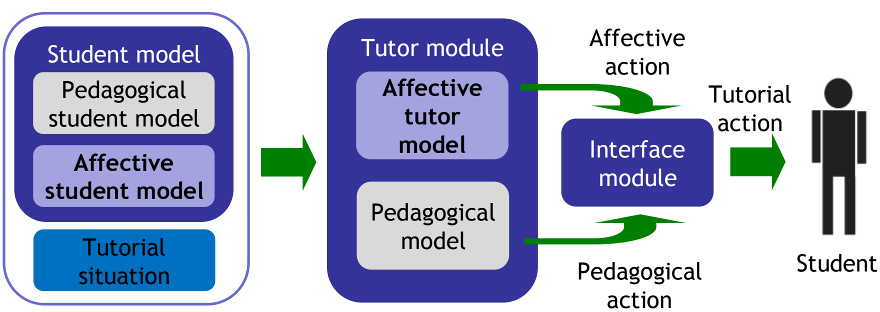


Fig. 1.2 General diagram for the affective behavior model. The model is composed of an affective student model and an affective tutor model. The tutor model produces an affective action, considering the affective and pedagogical student models and the tutorial situation. The affective action is a component of the tutorial action to be presented to the student

The ABM allows intelligent educational systems to make a mapping from the student's affective and pedagogical states to tutorial actions by means of the student model. In the next section, the affective student modeling is discussed.

1.4 The Affective Student Model

There are several proposals to predict or diagnose the individual's affect. These include facial expressions, and even, direct inquiry to the students as to their affective state. However, the latter is not a reliable means to ascertain affect, as asking them disrupts their concentration. People tend to be affable and to give a favorable answer even when the questioner is a computer (Reeves and Nass 1996).

Our affective student model uses the CME (Ortony et al. 1988) to provide a causal assessment of student's emotions based on contextual information. CME defines emotion as the end result of a cognitive appraisal of the current situation with respect to one's goals, principles and preferences. In this way, emotions represent a positive or a negative reaction, with respect to consequences of events, actions of agents and aspects of objects. Thus, an individual's emotions are related to the elements in the current situation: events, objects and agents, including him. Fig. 1.3 aims to show the fundamentals of the CME.

CME proposes 22 emotions and the emotions are classified according to the causes which elicit them: the consequences of events, the actions of agents and the aspects of objects. The elicited emotion also depends on the relevance of the event, agent or object to the individual; therefore, the model establishes parameters which represent the intensity of emotion.

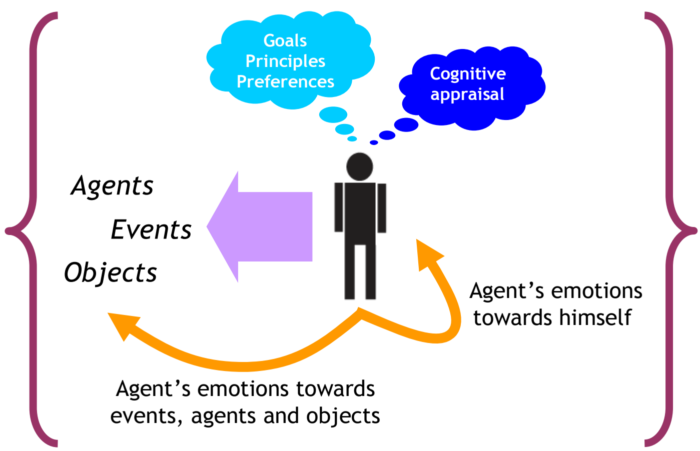


Fig. 1.3 Cognitive Model of Emotion basic diagram. The CME defines emotion as an end result of a cognitive appraisal of the current situation with respect to one's goals, principles and preferences. Emotions are elicited by elements included in the actual situation; they can be events, agents and objects

With regard to the consequences of events, in a tutorial session there are events that are pertinent to learning, such as an explanation (tutor's event) or the completion of an exercise (student's event). These events produce results that have an influence on the well-being of the student; therefore, these results trigger in the students the states we want to evaluate, such as *joy* and *distress*.

With reference to actions of agents, the tutorial situation contains two agents that are relevant for learning: the student and the tutor. These agents fulfill actions and the results of these actions cause emotions in the students, such as *pride* or *shame* if the student carried out the action; or *admiration* or *reproach* if the tutor performed the action. Thus, the results are attributable to the agent who carried out the action and consequently the student's emotions are focused on that agent.

We do not include emotions which emerge as a reaction to the aspect of objects, such as *love* and *hate*. Thus, from the set of emotions proposed by CME, the affective student model includes six emotions: *joy*, *distress*, *pride*, *shame, admiration* and *reproach*. The emotions *joy* and *distress* are reactions by the individual to an event in the tutorial session. The emotions *pride* and *shame* emerge as a consequence of the student's action. The emotions *admiration* and *reproach* emerge as a consequence of the tutor's action. Fig. 1.4 depicts how these emotions emerge in our model consistent with CME. The agent (student or tutor) performs an action and the student observes the result; he compares the results with his goal, causing emotions in keeping with the fulfillment of the student's goal.

Based on our affective student model on a comparison between the current situation and the individual's expectations, we make a prediction about the affective state. In that way, we do not need physical indicators such as facial expression, blood pressure, etc., or evidence of the individual's behavior for the affective state. Nevertheless, having additional indicators allows disambiguation of certain states; an approach to a student model with several indicators is given in (Conati and Maclaren 2009).

According to CME, goals are essential to determine the affective state. As in the case of understanding the student's affect, we believe goals cannot be explicitly asked of the student during the interaction; because in order the student to provide a reliable answer, he would need to understand the question and be introspective, and errors can occur. Consequently, the goals in our model are inferred from indirect sources of evidence; we use personality traits and student's knowledge as a predictor of the student's goals. We based the personality traits on the Five-Factor Model (Costa and McCrae 1992, Boeree 1998), which considers five dimensions of personality: *openness, conscientiousness, extraversion, agreeableness and neuroticism.* The Five-Factor Model describes each of these dimensions of personality and establishes their characteristics of behavior. For example, a person who has a high score in the *openness* dimension is a person willing to experience new things, is always disposed to dialogue, and has a high capacity for invention. Whereas, if he has a low score in *openness*; then he is a person with little disposition toward new experiences.

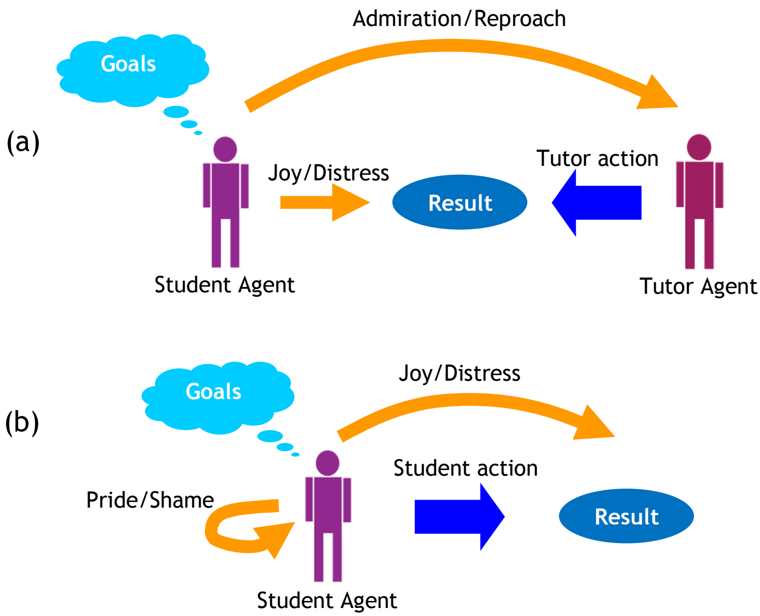


Fig. 1.4 Emotions represented in the affective student model. (a) Emotions from the student agent toward the tutorial session (joy/distress) and toward the tutor agent (admiration/reproach). These emotions surface when the student sees the result of the actions of the tutor. (b) Emotions of the student toward the tutorial session (joy/distress) and toward himself (pride/shame). These emotions are generated when the student observes the results of his actions. In accordance with the CME, the student compares the current situation against his goals

In a survey devoted to understand the relationship between personality and learning (Heinström 2000), it is stated that personality dimensions in terms of learning styles are reflected in learning strategies, and finally the personality produces the results in learning. This study also pointed out that the characteristics of personality act as guides for motivation and for learning strategies. The conclusions of the above study establish that the prominent dimensions for learning are *neuroticism* and *conscientiousness*. It also sets a relationship of learning with *openness*; but this relationship has not as yet been fully proven (Heinström 2000).

According to (Heinström 2000), the most important relationship of learning is with the *conscientiousness* dimension, since this personality dimension is related to discipline in one's work. Interest in the subject matter, concentration and the concept of study is easy. Students with this personality have intrinsic motivation and a positive attitude toward study. The *neuroticism* dimension is related to the lack of concentration, the fear of failure and the experience of studying as stressful. The neuroticism dimension is coupled with the lack of critical ability and difficulty in understanding the relationship between things. Students with this personality concentrate on memorizing, without interest in understanding and finding meaning in the material. In these cases the motivation of the students is extrinsic.

The *openness* dimension is related to questioning and analyzing arguments, critical evaluation, searches in the literature and building relationships. The students with this personality are analytical, logical, and they relate what they learn to their prior knowledge; their motivation is intrinsic and they seek personal and independent comprehension.

As a complementary way to understand students' goals, we use the student's current knowledge about the subject matter. We think knowledge has an effect on the absence or the presence of certain goals, especially on those goals related to learning, which is our main concern in this context.

As we mentioned, CME states that emotions emerge as a consequence of a cognitive appraisal of the current situation and goals; in our context we ascertain the current situation from the tutorial situation, i.e., the results of student actions (exercises, tests, etc.).

In that way, we can make a prediction about the affective state of the student based on contextual information; i.e., the current state of the student's knowledge, his personality traits and the tutorial situation. In the next section, we present the structure of the affective student model and we describe how it is built.

1.5 Building the Affective Student Model

As the process of establishing the affective state of students involves uncertainty, we rely on DBN for that task, due to their strong mechanisms for managing uncertainty. We use a DBN that probabilistically relates the student's personality, goals and interaction events with the student's affective states, based on the theory defined by CME. The dynamic network allows for the modeling of the changing nature of the affective state and representing the impact of the previous state in the current affective state. In our model, reaching the goals is the main factor influencing the affective state which in turn is influenced by the tutorial situation (the results of student's actions) and the student's goals. Consequently, the goals change during the tutorial session as the student learns. Fig. 1.5 shows a high level representation of the affective student model.

The affective state is not static but it changes over time as a result of the changing environment and the particular interpretation of the situation of each individual. The DBN models this dynamic nature of the affective state and its influence on the next state. In our model, the affective state changes after the student carries out an action. The dynamic network includes two time slots at any given time. A time slot is added and a time slot is discarded after each student's action. To infer the affective state at t_n we use the knowledge state of the student, the tutorial situation and the personality traits of the student; this is used to predict the affect at t_{n+1}.

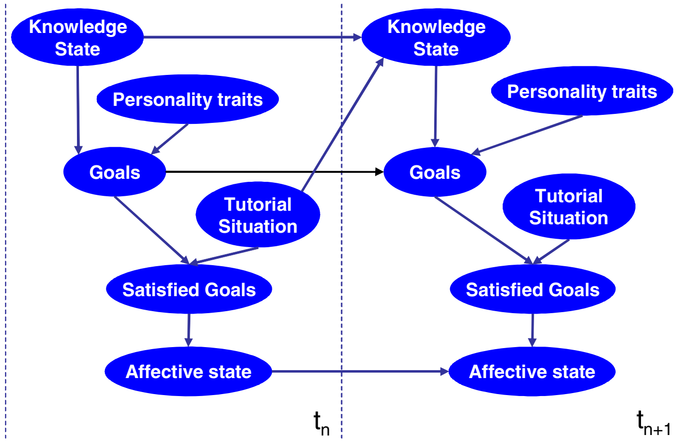


Fig. 1.5 High level DBN for the affective student model. We include two time slots to represent the dynamic behavior of affect and its impact in the next state

The student's appraisal of the current situation given his goal is represented by the relationship between the *goals* and the *tutorial situation* nodes via the *satisfied goals* node. The influence of the appraisal process on the student's affect is represented by the link between the *satisfied goals* node and the *affective state* node. This is our approach to the implementation of CME.

The goals change when the student interacts with the learning environment, that is, when the student is acquiring new knowledge. Another indicator for the goals is personality; however, this component does not change during the tutorial session. The influencing factors of goals are represented by knowledge state and personality traits nodes.

We call the network in Fig. 1.5 a high level representation of the model because of each node in that network is actually a set of nodes in the detailed model. Further, we describe comprehensively the DBN through a couple of subjects: how we build the affective student model and how we obtain the values of the nodes.

Fig. 1.6 shows the detailed DBN in a test case for robotics in which the students learn by carrying out experiments, such as setting up and guiding a mobile robot. One specific moment in time is depicted in this network. The dependency relationships in the DBN have been set based on the literature (Costa and McCrae 1992, Boeree 1998, Heinström 2000) and on insights from teachers and intuition.

The first node in the network is the knowledge node. The evidence for this node comes from the student action results (experiments, tests, etc.) by means of a pedagogical student model. The pedagogical student model is also a DBN that represents the current experiment and contains a node for each topic in the experiment. The probability of knowing each of these topics influences the probability of knowledge for the entire experiment. The knowledge node has two values: *knows* and *does not know*. This process is presented in Fig. 1.7.

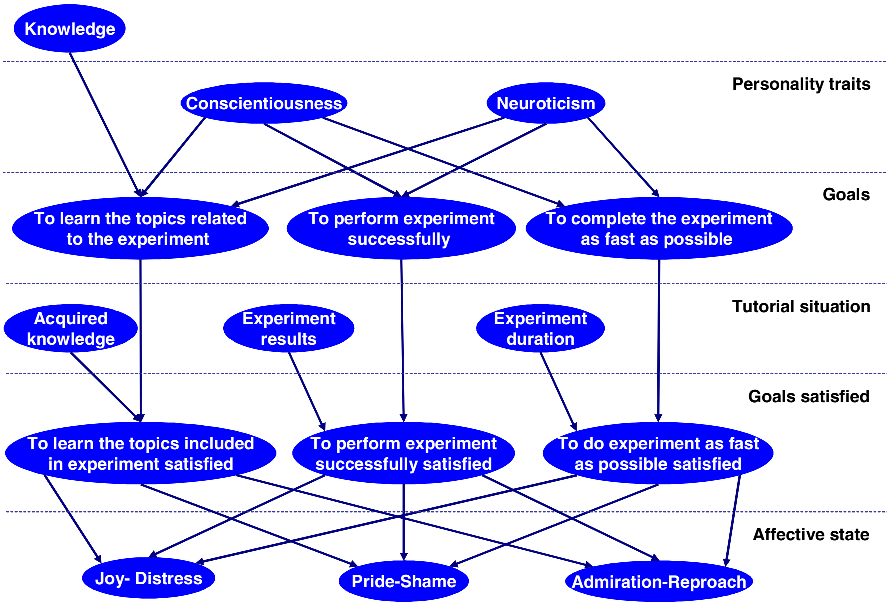


Fig. 1.6 Detailed affective student model represented by a DBN. Each set of nodes is a detailed representation of the DBN at a specific time

This DBN for the pedagogical student model is specific for a particular experiment. Each experiment in the learning environment has a different structure based on the main topic of the experiment. In this case, we show an experiment with four topics.

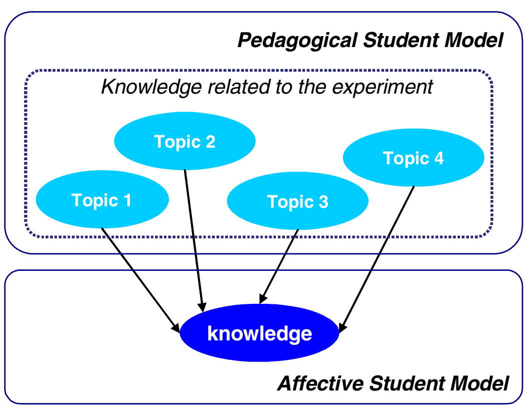


Fig. 1.7 Knowledge node. Evidence for this node is taken from the pedagogical student model, from the probability of knowing each topic in the experiment

The next set of nodes is the *personality* nodes. As previously indicated, for the personality traits of the student we use the neuroticism and conscientiousness dimensions. To obtain priors for these personality nodes, we conducted a study with 58 students. This group of students is a representative sample of the population who will use the learning environment given that they are graduate or undergraduate students, they are attending the same courses, and they are in the same age group. The complete survey can be found in (Hernández 2008). In this study, the students answered a personality test based on the Five-Factor model (Boeree 2005). The test consists of 50 adjectives, and the individuals have to rate how much the adjectives applies to them. The grade of the test indicates if the student is at a low, average or high level for each personality dimension. According to the survey, most of the students (78%) are at average level and a smaller student group (22%) is at low level for both personality dimensions. In this sample no one student is at high level for no one personality dimension. Coincidentally, both personality traits have the same percentages for the three personality levels. Based on this study, we establish the priors for the personality nodes presented in Table 1.1. In our DBN, the personality nodes have three values: high, average and low.

To have a more precise estimation of students' personality, they can answer the same personality test. To establish the dependency relationships in the DBN between personality nodes and goal nodes we considered the personality dimension description as stated by the Five-factor model (Costa and McCrae 1992, Heinström 2000, Boeree 1998). For example, if the student has a conscientious personality and limited understanding of the subject matter, the probability of having the goal *to learn the topics related to the experiment* is high, because he is a responsible person who cares about his performance. On the other hand, if the student is a neurotic person, there is a higher probability of having the goal *to perform the experiment successfully* rather than to learn, because a neurotic person wants to have immediate and tangible success.

The student's knowledge about the topics and the student's personality are accounted to infer the students' goals. We included three goals in the affective model: 1) to learn the topics related to the experiment; 2) to perform the experiment successfully; 3) to complete the experiment as fast as possible.

The reasons for establishing these goals are based on the nature of the task. That is, to perform an experiment to learn mobile robotics. The first goal can be present due to the main objective of the task: to complete an experiment for learning. The second goal can be present because of the student can wish to have success in reaching a target.

Table 1.1 Priors for *conscientiousness* and *neuroticism* personality nodes

Values	Conscientiousness	Neuroticism
1) High	0.01	0.01
2) Average	0.77	0.77
3) Low	0.22	0.22

The third goal can be present because generally students want a quick reward. In Table 1.2, we present the conditional probabilities table (CPT) for the goal *to perform experiment successfully* node. This node has two values: *present* and *absent*, and the influencing nodes, conscientiousness and neuroticism, have three values: *high*, *average* and *low*.

The CPT for the other two goals are similar to CPT in Table 1.2: the goal node has two influencing nodes, the personality nodes and its probabilities of having (present value) or not having the goal (absent value) are based on the personality traits. Additionally, the probabilities of the goal *to learn the topics related to the experiment* are based on the student's current knowledge.

The next set of nodes is the *tutorial situation* nodes (Fig. 1.6). The information for the *tutorial situation* nodes comes from the results of the student action by means of the pedagogical student model. We use the knowledge about the topics included in the experiment, and based on the specific experiment, data such as: how many times the student made a correction to the robot's track, if he reached or did not reach the target, and how long it took to reach to the target. This process is shown in Fig. 1.8.

Consequently, the student's appraisal of the current situation given his goal (CME) is represented by the relationship between the *goals* and *tutorial situation* nodes via the *satisfied goals* nodes.

Table 1.2 CPT for the to perform experiment successfully goal node

Goal 2: *to perform experiment successfully*									
Conscientiousness	High			Average			Low		
Neuroticism	High	Avg	Lw	High	Avg	Lw	High	Avg	Lw
Present	0.9	0.8	0.7	0.8	0.7	0.6	0.7	0.6	0.4
Absent	0.1	0.2	0.3	0.2	0.3	0.4	0.3	0.4	0.6

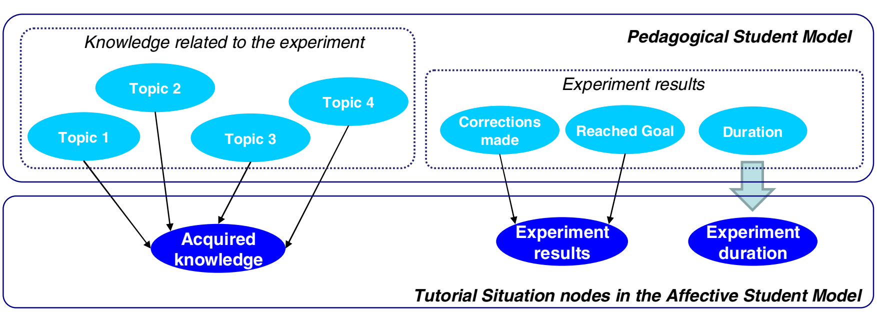


Fig. 1.8 Tutorial situation nodes. The tutorial situation nodes consider the pedagogical student model and the experiment's results

The probability that the goal has been satisfied depends on the presence or the absence of such a goal and the evidence of the results of the students' actions (tutorial nodes). However, it is more important to have the goal present in order to satisfy it. These nodes have two values: *satisfied* and *not satisfied*.

Finally, as stated by CME the emotions emerge as a comparison between goals and situation. In our model, the influence of the appraisal process on the student's affect is represented by the link between the *satisfied goals* nodes and the *affective state* nodes. We include in the model six emotions: joy, distress, pride, shame, admiration and reproach. These are represented as three pairs of mutually exclusive emotions: *joy-distress*, *pride-shame* and *admiration-reproach*. Each pair is represented by a binary node in the network. We used each pair of emotions as a dimension (see Fig. 1.9). We considered that for the same event/situation the student cannot have both emotions in the dimension; that is, the student cannot be happy and sad at the same time about the result of an experiment. It could be possible for several events; but it is not our case, we are evaluating the emotion toward one event only.

The *joy-distress* affective state node represents the emotions the student can have regarding the situation. That is: he is happy because of he learned, or due to he got the target, or because of he completed the experiment quickly. The *pride-shame* affective state node represents the emotions from the student towards himself. It means, he is proud of himself because of he learned the topics in the experiment, or due to he completed the experiment successfully, or because he achieved the goal quickly. The *admiration-reproach* affective state node represents the emotion from the student towards the tutor depicted by an animated agent (as a part of our study we included in the learning environment an animated agent to present the instruction and to be the face of the tutor). The student can feel admiration for the tutor as a result of the tutor taught him and therefore he reached his goals. In Table 1.3, we show the conditional probabilities for the *joy-distress* node.

Fig. 1.9 Emotion dimensions. The affective student model includes three pairs of mutually exclusive emotion. This consideration applies only to a same event

Table 1.3 Conditional probabilities table for the joy-distress node

Joy-distress								
Goal to learn the topics related to the experiment satisfied	Yes				No			
Goal to perform experiment successfully satisfied	Yes		No		Yes		No	
Goal to complete the experiment as fast as possible satisfied	Yes	No	Yes	No	Yes	No	Yes	No
Joy	0.9	0.8	0.7	0.8	0.7	0.6	0.7	0.6
Distress	0.1	0.2	0.3	0.2	0.3	0.4	0.3	0.4

In the following section, we describe how we evaluated the affective student model we just presented.

1.6 Evaluation of the Model

In order to evaluate the affective student model we integrate the ABM into an intelligent learning environment to learn mobile robotics (Noguez and Sucar 2005). In this environment the students learn by carrying out experiments about to set up and guide a mobile robot to reach the target. Once they completed the experiments, they learn a lesson based on their performance. The instruction is based on a probabilistic representation of the students' knowledge state. The pedagogical actions are explanations about the topics in the current experiment. This learning environment presents the instruction by means of a textual explanation without an agent or face for the tutor. However, in our affective student model we assess the student's emotion toward the tutor. Therefore, we need a face for the tutor so that when we evaluate the model we can ask for the emotion toward the tutor without causing confusion in the student. For that reason, we integrate an animated agent into the learning environment.

To include a suitable agent, we conducted a survey in which we asked nine teachers to select an animated character and appropriate animations to be integrated into the intelligent environment. Teachers were presented with the possible animations displayed by four characters of Microsoft Agent® (Microsoft 2005), so they can see the potential of the every animated character. In Fig.1.10, the characters of Microsoft Agent® presented to teachers are shown, left to right: Robbie, Genie, Peedy and Merlin.

Fig. 1.10 Characters of Microsoft Agent. Teachers could see the potential of these characters and select the one they considered suitable for the intelligent environment audience

Teachers could select a character and see all of its animations as many times as they wanted. The selected animations were included in the ABM as affective actions. The character Robbie was selected by seven teachers and the character Merlin was selected by two teachers. Even the teachers that selected Robbie acknowledged Merlin is much more expressive than Robbie; they thought Robbie was more suitable for the domain (Robotics) and for the students' ages (college students).

To evaluate the performance of the affective student model, we conducted a Wizard of Oz study with a group of 20 students. This sample is small but it is representative of the type of students who will be using the learning environment. The aim of a Wizard of Oz study is to obtain information for designing and evaluating prototypes or systems which have not yet been finished (Dow and MacIntyre 2007, Dow et al. 2005). This type of study has been used since human-computer interaction began and it has been widely used to emulate technologies in interactive systems (Dow and MacIntyre 2007, Anderson et al. 2002). It consists of employing operators or mechanisms hidden from the user temporarily to emulate unfinished components of a computer system during its development (Dow et al. 2005). In our case we did not have the Microsoft Agent® completely integrated into the intelligent environment for learning robotics. Therefore, we videotaped several tutorial scenarios, and for every scenario we showed the animation (affective action) according to the affective behavior model.

Aside from having personality priors, we requested the participating students answer the same personality test before using the system in order to have a more precise evaluation of personality. As the first point in the survey, the students answered a personality test based on the Five-Factor model (Boeree 2005). This test is the same test used to obtain the priors for the personality nodes. It is composed of 50 adjectives such as talkative, sympathetic, envious, deep, careless, relaxed, average, bold, kind and moody. The students have to rate how each adjective applies to them.

The survey consisted of, presenting to the students, three different tutorial scenarios. The tutorial scenario included the development of an experiment and a tutorial action presented by the animated character Robbie. The tutorial action was selected considering the student's affective state and the tutorial situation presented in the scenario. After presenting each tutorial scenario, the students were asked about their affective state given the tutorial situation with the purpose of comparison to the affective state established by the affective student model. In Fig. 1.11, we present a block diagram of the Wizard of Oz study.

1.7 Results

We compared the affective state reported by the students with the affective state established by the affective student model. The results are summarized in Table 1.4. We found that the model estimated the affective state correctly: for the emotion *joy-distress* in 72% of the cases, for the emotion *pride-shame* in 70% of the cases and for the emotion *admiration-reproach* in 55% of the cases. As we can see, the model reached a high precision for the emotions *joy-distress* and *pride-shame*. However for the emotion *admiration-reproach* the precision of the model is not so high.

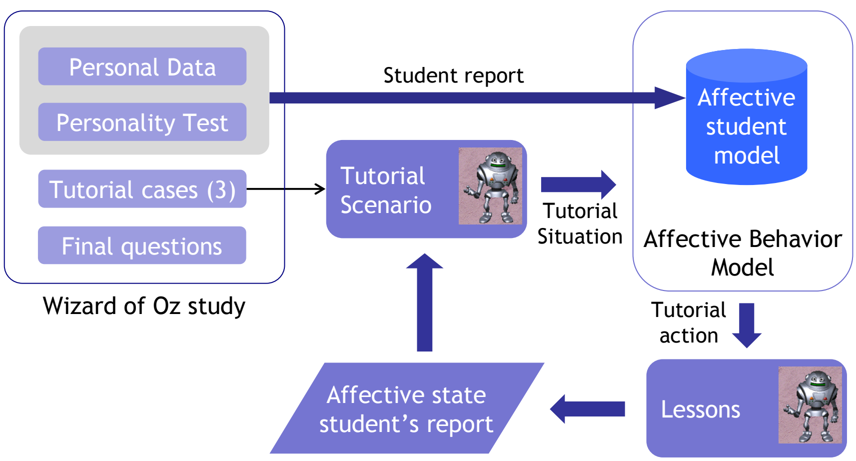


Fig. 1.11 Block diagram for Wizard of Oz study to evaluate the affective student model. The students were presented with three tutorial scenarios and they were asked about their affect after each tutorial action; their responses were compared to the affective state established by the affective student model

Table 1.4 Percentage of agreement between the affective state established by the affective student model and the affective state reported by the students

	Joy-distress	Admiration-reproach	Pride-shame
1) Agreement	43	33	42
2) Disagreement	17	27	18
Percentage of agreement	72%	55%	70%

We suppose: the emotions from students toward teachers evolve slowly. We think: students believe they are learning on their own and, in general, they do not think that teachers are instructing them well. This concurs with comments of teachers who rated the emotions from the students toward them as being mostly negative (the survey is reported in detail in Hernández 2008). It is also possible that the students did not have enough information to evaluate this emotion because of they did not receive a knowledge test. We have to conduct further investigations to validate this hypothesis and refine the affective student model and the ABM.

In this chapter, we have presented an evaluation of the affective student model, a component of the ABM. Moreover, we have tested and evaluated the complete affective behavior model in another domain. We have also integrated it into an educational game to learn number factorization (Manske and Conati 2005). We carried out a controlled user study with 82 actual students. The trial held a control group and an experimental group. The control group used the system without the ABM and the experimental group interacted with the system with the ABM. The students were given a pre-test and a post-test in order to establish learning gains by using the system. The learning gains of the groups were compared using the statistical *t*-student test. The results of the studies show positive impact on students' learning when the affective behavior model is incorporated, as shown in Table 1.5. The complete survey is presented in (Hernández et al. 2009b).

Table 1.5 Statistical analysis of the learning gains in each group, control and experimental, and between both groups

Grade	Control Group Pre-test/post-test		Exp. Group Pre-test/post-test		Learning gains Cntl grp/Exp grp	
	t	P (1-tailed)	t	P (1-tailed)	t	P (1-tailed)
6° gr.	2.55	0.09	6.95	0.000036	8.10	0.04
7° gr.	0.29	0.80	0.70	0.210000	0.36	0.69
8° gr.	0.63	0.58	0.53	0.370000	0.09	0.93
9° gr.	1.10	0.08	0.19	0.800000	0.97	0.28

1.8 Discussion

The results of our investigations about developing an intelligent representation of the student affect are encouraging as they show high agreement between the reports of the students and the results of our affective student model. Additionally, we also get positive results in the evaluation of the complete ABM due to high learning gains when we used the affective student model to instruct the student.

We decided to base our investigation on theoretical models of emotions and on indirect sources of evidence, such as personality, goals and results. This is because of we have tried to build an approach which does not interfere with the students' main task. Nevertheless, we pursue to deal with the lack of direct sources of evidence, such as biological signals, through the use of DBN. In addition, we assert: Bayesian reasoning provides strong mechanisms to work with any minimal evidence in order to manage the inherent uncertainty in the assessment of both the current relevant student affective states and the effects of the tutor's actions on them.

The ABM allows intelligent educational systems to make a mapping of a student's affective and pedagogical states to tutorial actions. Having two test domains with positive results suggest us that the ABM can be integrated into any intelligent learning environment. However, these results are not conclusive, as we need to include knowledge tests in order to prove whether our model helps students to learn. Nevertheless, the results that we have obtained so far will allow us to refine our models and to design other studies, and to progressively achieve a comprehensive approach to affective behavior.

1.9 Conclusions and Future Work

We have developed an affective behavior model for intelligent learning environments. The affective behavior model integrates an affective student model and an affective tutor model. In this chapter, we presented the first component, the affective student model, which was built considering the theory stated by the CME. We also presented the affective student model's evaluation in the mobile robotics domain via a Wizard of Oz experiment.

The results are encouraging, as they show strong agreement between the affective states given by the model with those of the students. The next step is to complete the integration of the affective student model and the complete ABM with the intelligent learning environment for mobile robotics. Subsequently, we aim to conduct a controlled user study and in this way, try to confirm our hypothesis: the learning process is improved when the affective state is considered.

Acknowledgments. We would like to thank Cristina Conati from the University of British Columbia, for many useful discussions on the definition of the affective model. We would also like to thank Amelia Hakim and Marisa Velasco Feld and the anonymous reviewers whose comments helped to improve this manuscript.

References

Abbasi, A.R., Uno, T., Dailey, M.N., Afzulpurkar, N.V.: Towards Knowledge-Based Affective Interaction: Situational Interpretation of Affect. In: Paiva, A., Prada, R., Picard, R.W. (eds.) ACII 2007. LNCS, vol. 4738, pp. 452–463. Springer, Heidelberg (2007)

Alexander, S., Hill, S., Sarrafzadeh, A.: How do human tutors adapt to affective state? In: Proceedings of Adapting the Interaction Style to Affective Factors Workshop of UM (2005)

Anderson, G., Hook, K., Mourao, D., Paiva, A., Costa, M.: Using a Wizard of Oz study to inform the design of SenToy. In: Mackay, A., Amowitz, J., Gaver, W., Verplank, B., Sutcliffe, A. (eds.) Proceedings of DIS, pp. 349–355. ACM Press, London (2002)

Boeree, G.: Big Five Minitest, `http://www.ship.edu/~cgboeree/` (accessed August 15, 2005)

Boeree, G.: Teorías de la personalidad. E-book. Psychology Department of Shippensburg University (1998), `http://www.ship.edu/~gboeree/personalidad.html` (accessed August 15, 2005)

Burns, H.L., Capps, C.G.: Foundations of intelligent tutoring systems: An introduction. In: Polson, M.C., Richardson, J.J. (eds.) Foundations of Intelligent Tutoring Systems, pp. 1–20. Lawrence Erlbaum Associates Publishers, Hillsdale (1988)

Conati, C., Maclaren, H.: Empirically building and evaluating a probabilistic model of user affect. User Modeling and User-Adapted Interaction 19(3), 267–303 (2009)

Costa, P.T., McCrae, R.R.: Four ways five factors are basic. Personality and Individual Differences 13(1), 653–665 (1992)

Dow, S., MacIntyre, B.: New media collaboration through Wizard-of-Oz simulations. In: Proceedings of HCI and New Media Arts: Methodology and Evaluation Workshop of CHI (2007)

Dow, S., MacIntyre, B., Lee, J., Oezbek, C., Bolter, J.D., Gandy, M.: Wizard of Oz support throughout an iterative design process. IEEE Pervasive Computing: Special Issue on Rapid Prototyping 4(1), 18–26 (2005)

D'Mello, S., Taylor, R., Davidson, K., Graesser, A.: Self Versus Teacher Judgments of Learner Emotions During a Tutoring Session with AutoTutor. In: Woolf, B.P., Aïmeur, E., Nkambou, R., Lajoie, S. (eds.) ITS 2008. LNCS, vol. 5091, pp. 9–18. Springer, Heidelberg (2008)

Dragon, T., Arroyo, I., Woolf, B.P., Burleson, W., el Kaliouby, R., Eydgahi, H.: Viewing Student Affect and Learning through Classroom Observation and Physical Sensors. In: Woolf, B.P., Aïmeur, E., Nkambou, R., Lajoie, S. (eds.) ITS 2008. LNCS, vol. 5091, pp. 29–39. Springer, Heidelberg (2008)

Faivre, J., Nkambou, R., Frasson, C.: Towards empathetic agents in tutoring systems. In: Russell, I., Haller, S. (eds.) Proceedings of FLAIRS, pp. 161–165. AAAI Press, Florida (2003)

Heinström, J.: The impact of personality and approaches to learning on information behavior. Information Research: an International Electronic Journal 5(3) (2000), `http://informationr.net/ir/5-3/paper78.html` (accessed August 15, 2005)

Hernández, Y.: Modelo de comportamiento afectivo para sistemas tutores inteligentes. PhD dissertation, Tecnológico de Monterrey (2008)

Hernández, Y., Sucar, L.E., Arroyo, G.: Obtaining teachers' expertise to refine an affective model in an intelligent tutor for learning robotics. In: Hernández, A., Monroy, R., Reyes, C.A. (eds.) Proceedings of MICAI, pp. 122–127. IEEE Press, Los Alamitos (2009)

Hernández, Y., Sucar, L.E., Conati, C.: Incorporating an affective behavior model into an educational game. In: Lane, C.H., Guesgen, H.W. (eds.) Proceedings of FLAIRS, pp. 448–453. AAAI Press, Florida (2009b)

Jaques, P.A., Viccari, R.M.: Considering student's emotions in computer-mediated learning environments. In: Zongmin, M. (ed.) Web-Based Intelligent E-Learning Systems: Technologies and Applications, pp. 122–138. Information Science Publishing, Hershey (2005)

Johnson, W.L., Rickel, J.W., Lester, J.C.: Animated pedagogical agents: Face-to-face interaction in interactive learning environments. International Journal of Artificial Intelligence in Education 11(1), 47–78 (2000)

Kim, Y., Xu, B., Wei, Q.: Virtual peers scaffold motivation to learn. In: Proceedings of Modeling and Scaffolding Affective Experiences to Impact Learning Workshop of AIED (2007)

Kort, B., Reilly, R., Picard, R.W.: An affective model of interplay between emotions and learning: Reengineering educational pedagogy – Building a learning companion. In: Okamoto, T., Hartley, R., Kinshuk, Klus, J. (eds.) Proceedings of ICALT, pp. 43–46. IEEE Press, Madison (2001)

Lehman, B., Matthews, M., D'Mello, S., Person, N.: All alone with your emotions: an analysis of student emotions during effortful problem solving activities. In: Woolf, B.P., Aimeur, E., Nkambou, R., Lajoie, S. (eds.), Proceedings of ITS. LNCS, vol. 5091, pp. 9-18. Heidelberg: Springer (2008)

Manske, M., Conati, C.: Modelling learning in educational games. In: Looi, C., McCalla, G.I., Bredeweg, B., Breuker, J. (eds.) Proceedings of AIED, pp. 411–418. IOS Press, The Netherlands (2005)

Microsoft Corporation. Microsoft Agent (2005), `http://www.microsoft.com/msagent/default.asp` (accessed November 1, 2005)

Murray, R.C., VanLehn, K.: DT Tutor: A Decision-Theoretic, Dynamic Approach for Optimal Selection of Tutorial Actions. In: Gauthier, G., VanLehn, K., Frasson, C. (eds.) ITS 2000. LNCS, vol. 1839, pp. 153–162. Springer, Heidelberg (2000)

Noguez, J., Sucar, L.E.: A Semi-open Learning Environment for Virtual Laboratories. In: Gelbukh, A., de Albornoz, Á., Terashima-Marín, H. (eds.) MICAI 2005. LNCS (LNAI), vol. 3789, pp. 1185–1194. Springer, Heidelberg (2005)

Ortony, A., Clore, G.L., Collins, A.: The cognitive structure of emotions. Cambridge University Press, New York (1988)

Picard, R.W.: Affective computing. MIT Press, Cambridge (2000)

Qu, L., Wang, N., Johnson, W.L.: Using Learner Focus of Attention to Detect Learner Motivation Factors. In: Ardissono, L., Brna, P., Mitrović, A. (eds.) UM 2005. LNCS (LNAI), vol. 3538, pp. 70–73. Springer, Heidelberg (2005)

Reeves, B., Nass, C.: The media equation: How people treat computers, television and new media like real people and places. Cambridge University Press, New York (1996)

Teeters, A., El Kaliouby, R., Picard, R.W.: Self-Cam: Feedback from what would be your social partner. In: Proceedings of SIGGRAPH, p. 138. ACM Press, Boston (2006)

Zakharov, K., Mitrovic, A., Johnston, L.: Towards Emotionally-Intelligent Pedagogical Agents. In: Woolf, B.P., Aïmeur, E., Nkambou, R., Lajoie, S. (eds.) ITS 2008. LNCS, vol. 5091, pp. 19–28. Springer, Heidelberg (2008)

Abbreviations

ABM	Affective Behavior Model
AVG	Average
CME	Cognitive Model of Emotions
CPT	Conditional Probabilities Table
DBN	Decision Bayesian Network
EXP	Experimental
GR	Grade
GRP	Group
LW	Low

Chapter 2
ALEM: A Reference Model for Educational Adaptive Web Applications

Mohammed Tadlaoui[1], Azeddine Chikh[2], and Karim Bouamrane[3]

[1] Ecole Preparatoire Sciences et Techniques, BP165 RP Bel Horizon, Tlemcen, Algeria
mtadlaoui@hotmail.com

[2] College of Computer and Information Sciences, King Saud University, P.O. BOX 51178, Riyadh 11543, Kingdom of Saudi Arabia
az_chikh@ksu.edu.sa

[3] Departement d'informatique, Oran Senia University, BP 1524 EL Mnaouer, Oran, Algeria
bouamrane.karim@univ-oran.dz

Abstract. The adaptive hypermedia systems or adaptive web applications is a research area between hypermedia and user modeling. It can customize hyperspace to different users. The existing reference models are generic and are not dedicated to educational systems. This chapter presents in the first part, a reference model that is specific to adaptive educational hypermedia systems. This model is called ALEM (Adaptive Learning Environment Model). It consists of a domain model, a learner model, a course structuring model and an adaptation model. The main contribution of this model is modeling the adaptive learning curriculum. Furthermore, we develop the UML Tutor application which is an educational adaptive hypermedia system based on our reference model.

2.1 Introduction

Adaptive hypermedia systems (AHS) is an area of research that tries to provide the user with content adapted to his needs. AHS are used in several application domains such as educational systems, information systems, online help systems and online systems for information retrieval.

AHS is a set of nodes and links that allow a user to navigate in the structure of the hyperspace and dynamically customize the various visual aspects of hypermedia to the user's needs. Two types of adaptation exist (De Bra 2008): content adaptation and link adaptation. The first type is used to display and adjust the content of the pages to the characteristics and needs of the user. The second type allows to customize and limit the possibilities of navigation in hypermedia.

There are several methods to implement these two types of adaptation such as the comparative explanation or the additional explanations for the content adaptation and annotation, sorting or link hiding for link adaptation.

A. Peña-Ayala (Ed.): Intelligent and Adaptive ELS, SIST 17, pp. 25–48.
springerlink.com

A more recent taxonomy of adaptation methods and techniques can be found in (Knutov 2009). It distinguishes between content adaptation and presentation adaptation. Three models are used to adapt the hypermedia to the user's needs. These models are: 1) the domain model which is a representation of the subject of hypermedia through concepts and links between them; 2) the user model that represents user characteristics and needs; 3) the adaptation model that contains the rules for adaptation. In AHS, the user model is named learner model.

(Brusilovsky and Millan 2007) distinguish two types of user models: Models that represent the characteristics of the user as the knowledge, interests or goals and models that represent the work context of the user such as location or platform of the user. The first models are important to all adaptive web systems, while the latter are mainly used for adaptive mobile systems.

The user model may be a part of the AHS or it may be shared with multiple systems. In the latter case, we speak about user modeling servers. This type of servers is used in distributed environments where multiple adaptive systems access to this server to query or update user information.

This paper is particularly interested in the adaptation of the course plan to the learner specificities. A course plan is a path that a learner may take to meet a goal of a given training. In other words, a course plan is an ordered set of training resources that a learner must perform to reach his goal.

This chapter is organized as follows: Sect. 2.2 gives an overview of the most known reference models describing the AHS. Sect. 2.3 presents the reference model ALEM that we propose to model the educational AHS. Sect. 2.4 describes a prototype of the ALEM reference model which is called UML Tutor. The aim is to describe the various application modules that interact to satisfy the requirements of the ALEM reference model. This chapter will conclude with a presentation of possible extensions of our model and future work.

2.2 Adaptive Hypermedia Reference Models

The reference models describing the classic hypermedia systems (not adaptive) have begun to appear before the existence of the web, including the model Hypertext Abstract Machine (HAM) (Campbell and Goodman 1988). After the appearance of this model, others have followed and the best known of them is the Dexter Hypertext Reference Model (Halasz and Schwartz 1994).

The theoretical framework of the AHS explained in the previous section has served to define some reference models. The objectives of these models are:

- Model the existing AHS,
- Provide a platform for describing existing systems and specify future systems,
- Provide a platform to compare the different existing systems,
- Describe the basic concepts of AHS and the relationship between them,
- Separate the content, structure and presentation aspects of hypermedia systems.

Among the existing reference models of adaptive hypermedia systems, we found the seminal work about INSPIRE system (Papanikolaou et al. 2002), AHAM model (Wu 2002), Munich model (Koch and Wirsing 2002) and Social LAOS (Ghali and Cristea 2009). The third reference model is explained below.

2.2.1 The Munich Model

The Munich reference model is a model based on the Dexter model. It was developed independently of the AHAM model. The main contribution of this model is that it uses a graphical language for describing the different components of AHS. The layered architecture of Dexter model has been replaced by a Unified Modeling Language (UML) (OMG 2010) package diagram and the description of the user model, domain model and adaptation model has been illustrated by UML class diagrams. These diagrams are also used to describe different functions that are offered by the three models. Other than the graphical modeling, the Munich model makes the following extensions from its two predecessors:

- The components of the domain model are not only connected by navigational relationships (links), but also by other conceptual relations such as "part of", "prerequisite" and "variant of",
- The user model includes a user manager and a model for each user of the system composed of attributes and values,
- Two types of user attributes are taken into account: the attributes that are domain dependent and domain independent,
- The rules are classified into construction rules, acquisition rules and adaptation rules (content, link and presentation adaptation),
- The adaptation model also models the user behavior (browsing, input and user inactivity).

As AHAM, the Munich reference model is also a model which is not only used for educational adaptive hypermedia systems, but also for other types of AHS.

2.3 The ALEM Model

The weaknesses of some existing reference models and the limitation of others demonstrate the contribution that may exist in the proposed reference model. The ALEM model (Adaptive Learning Environment Model) (Tadlaoui et al. 2010) is an extension of the Munich Reference Model.

The greatest contribution of our model over existing models is the modeling of the course and the learning curriculum. The main objectives that have guided us for the development of this model are:

1. Describe existing and future adaptive educational hypermedia systems,
2. Include the concept of educational activity and the concept of educational curriculum,
3. Take into account all types of representation of the user model (overlay model, perturbation model, stereotyped model, etc.),
4. Model the goals of the learner and distinguish them from the goals of domain concepts.

The ALEM model is described using UML notation. This language has allowed us to perform a visual, rich and intuitive description of our model. It was also useful to show the concepts of our model and the various relationships between them. Items added to our model over the Munich model are marked on next figures with an asterisk (*).

The architecture of the ALEM model contains the same three layers existing in the Munich model, but it extends their functionality to better take into account the modeling of educational systems. In addition to these three layers, we added an educational layer.

The different layers of the model are shown in Fig. 2.1:

- The Within Component Layer contains the content and the structure of the hypermedia nodes and it also serves to separate the other layers from detail specific to media,
- The Storage Layer stores information on the structure of the hypermedia. This layer is composed of three models:
  - The domain model describing the scope of the hypermedia,
  - The learner model describing the learner characteristics useful for customizing the hyperspace,
  - The adaptation model describing the adaptation strategies and adaptation rules.
- The Educational Layer is an abstract representation of the course. This layer contains the structural model of the course,
- The Run-Time Layer is the description of how the nodes are presented in the front-end. This layer is responsible for interaction with the learner, acquisition of the learner data and management of sessions.

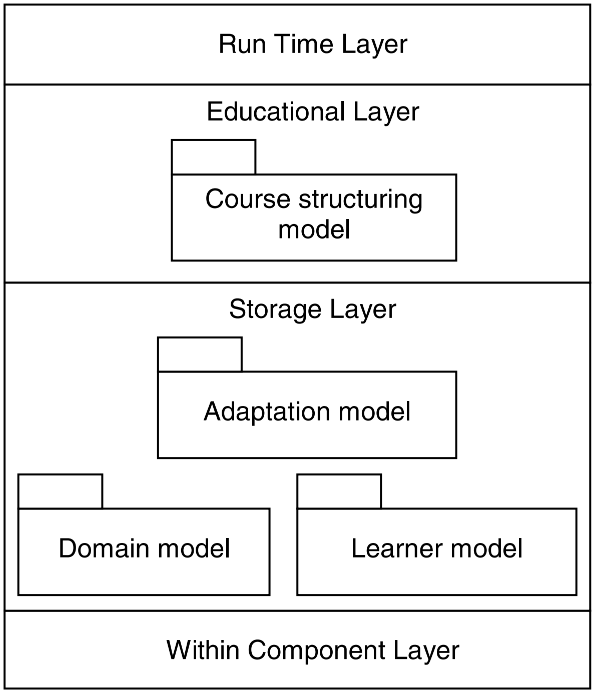


Fig. 2.1 The structure of the ALEM model

2.3.1 Domain Model

The domain model describes the structure of hypermedia as a set of components. Fig. 2.2 shows the domain model and its repartition in the storage layer and within component layer.

The application domain of the hypermedia is modeled by the class "Domain". It is described by attributes which permit to expose the definition of the application domain to other adaptive hypermedia for interoperability purposes. A domain is composed of a finite set of components.

The "Component" class is used to represent abstractly all components of the application domain: concepts, pages, fragments, goals and relationships between components. A component can be described by several descriptors using the Learning Object Metadata (LOM) formalism (IEE LTCS 2010) which is a standard that provides a set of attributes for describing learning objects.

The domain model can also describe, through the class "Presentation specification", how to present a component or relationship to the end user.

A concept is an abstract representation of the application domain information. It is defined by one or more pages.

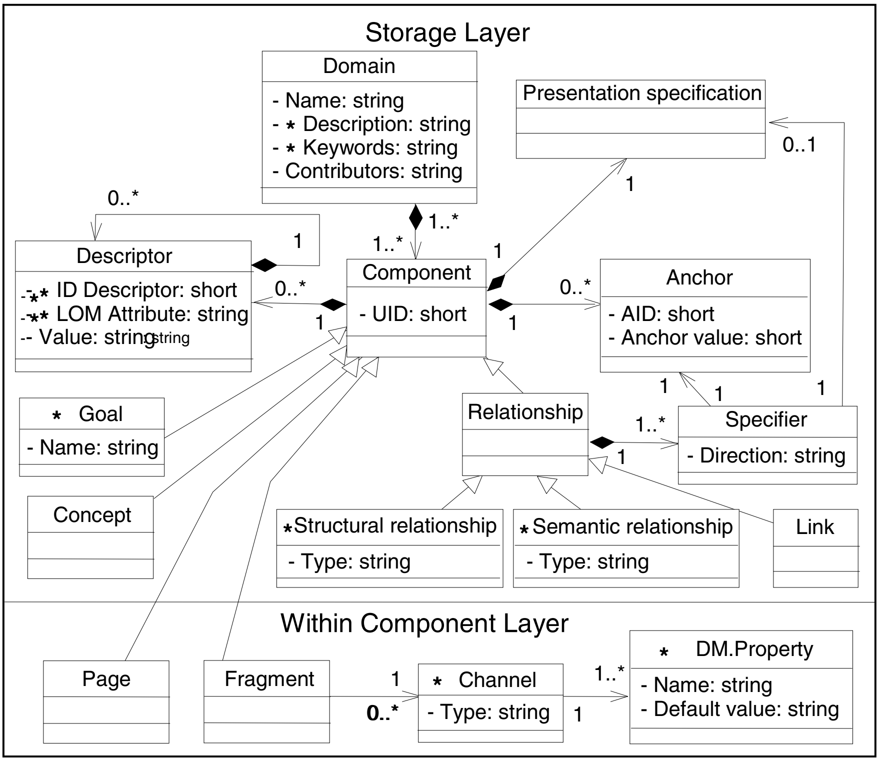


Fig. 2.2 Domain model

The page and fragment components are contained in the within component layer because they represent the content part of the hypermedia. A page consists of one or more fragments. A fragment belongs to a media channel (audio, video, Image, text, etc.). Each channel is described by properties (audio volume, text style, brightness of images, etc.) which are used to personalize the presentation.

The class "Goal" represents the objectives for which the component is created. A goal can be achieved by itself or by other goals. The reference model provides the possibility to define a hierarchy of goals associated with *n*-ary relationship of type “and” and “or”.

The component relationship is a link mechanism between various components of the domain. As shown in the model, a relationship can be either:

- A navigation link: It is the link that allows the user to move from one page to another to browse the hypermedia. This type of relationship connects the pages and fragments,
- A semantic relationship: It is used to express any kind of semantic links (prerequisite of, is similar to, is a version of, and, or, before, after, is important in, etc.) to connect all types of components,

- A structural relationship: It is used to express a composition relationship between concepts, pages and fragments. Among the possible structural relationships there is: is a, is part of and defines the concept.

A relationship may contain one or more specifiers to allow the description of the reflexive, binary or *n*-ary relationship. Each specifier is pointing to an anchor of a component. For example, in the case where the relationship is an hyperlink, the relationship must be composed of two specifiers. One point to the anchor of the source page and the other to the anchor of the destination page. In the source specifier, the value of the direction attribute is set to "From", and the other to "To". The value of the destination anchor is the URL of the destination page. By this mechanism of anchors, the most complex relationships can be modeled.

2.3.2 Learner Model

The learner model describes the learner by an identifier (LID) and a set of attributes. With these attributes the adaptive hypermedia system can represent the characteristics that are relevant for the application. We can distinguish several types of information contained in a learner model: name, background, experience, goals… classified in seven categories. The values assigned to attributes represent what the system believes about the learner. The learner characteristics are given below:

- Personal information: It is about information regarding the learner, such as: name, age, language, educational level, diplomas, certificates, etc.,
- Domain dependent knowledge: It is the knowledge that the user has acquired about a concept of the application domain of the hypermedia. It can be an exact value that a learner has about concept or a probability that a learner knows a particular concept,
- Domain independent knowledge: It is the knowledge in domains related to the domain of hypermedia, which are relevant for adaptation,
- Purpose: It represents the goal to be achieved by the learner. The class "Purpose" holds a time attribute to represent the time to achieve the purpose,
- Physical preference: It is related to the channel of media (audio volume, font, video speed, etc.),
- Cognitive characteristics: They are:
  - Cognitive capacity for example the speed of learning,
  - Cognitive preference, such as the type of interactivity with the system (active or passive), the density of content, the degree of difficulty, the resource type (formal, graphical, simulation, etc.).

The ALEM model allows taking into account the notion of stereotypes. These stereotypes have features with default values that are used principally in the initialization values of the characteristics of the learner. This model and its relationship between classes "Component" and "Property" is illustrated in Fig. 2.3.

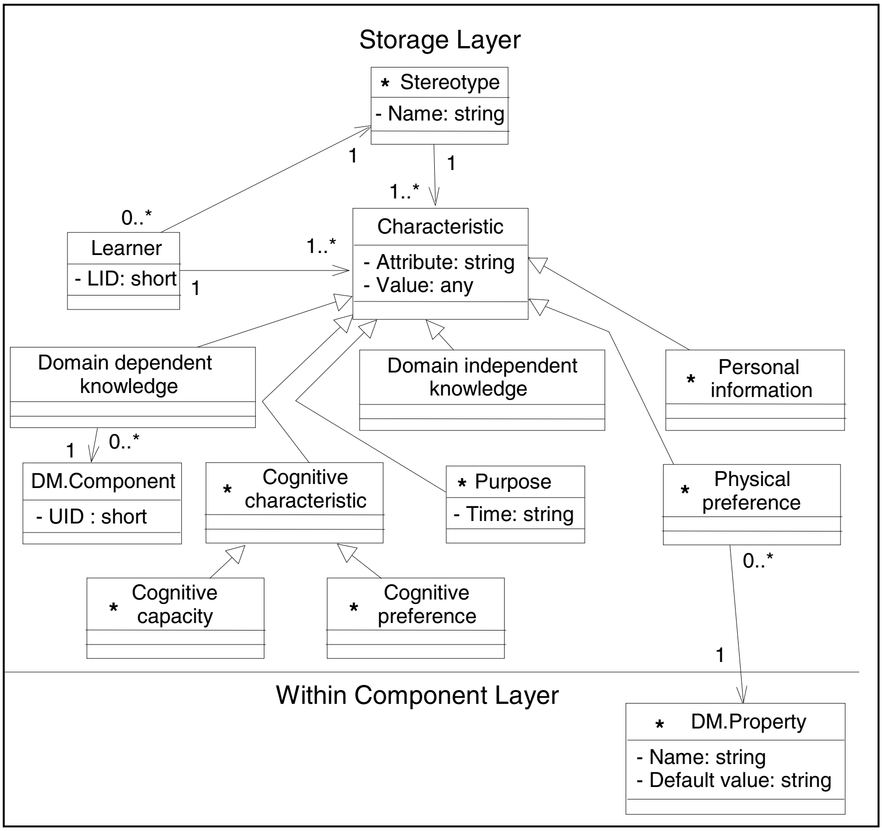


Fig. 2.3 Learner model

2.3.3 Adaptation Model

The adaptation model of ALEM, shown in Fig. 2.4, describes how the adaptation of link and content is made and how the learner model is updated. Adaptation is done using information from the domain model, the learner model and the learner interaction. The adaptation operation is performed by the adaptation engine. The basic element used for adaptation is the rule that determines how the pages are constructed and how they are presented to the learner.

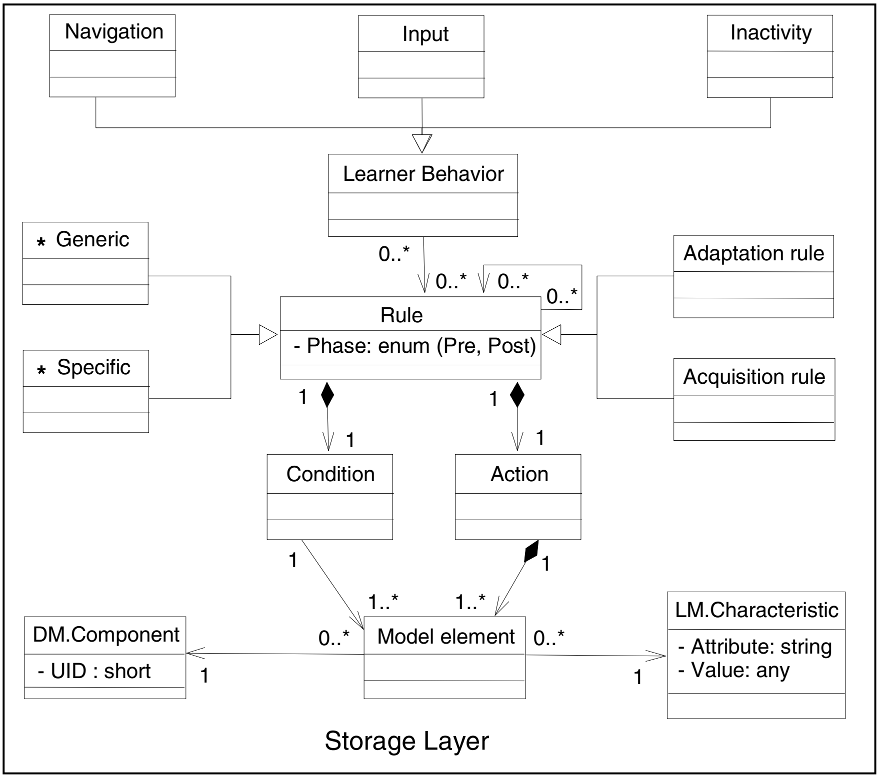


Fig. 2.4 Adaptation model

A rule consists of two parts:

- A condition necessary for the application of a rule,
- An action resulting from a rule. It can be the update of the learner model or the adaptation of the content and presentation.

The two parts of a rule contain expressions which are composed of elements and logical operators. These elements are mainly related to a characteristic of a learner or a component.

A rule can be applied before or after the page generation, following to the attribute phase of the class "Rule" which can take as value "post" or "pre".

A rule can belong to one of the following classes:

- Adaptation rule: It is used to adapt the content and links to the application,
- Acquisition rule: It is used to update the learner model.

An AHS may have predefined adaptation rules (class "Generic"). If these rules are not enough, other new rules (class "Specific") can be defined. The creation of a generic adaptation rule is made by the system designer, while a specific rule is created by the authors of the hypermedia system.

The behavior of the learner, embedded in the adaptation model, is classified according to the actions of the learner: navigation, input and inactivity. A rule is triggered either by the behavior of the learner or by another rule. Our model represents the rules in general and it is not a representation formalism of rules. The syntax of permitted rules depends on the hypermedia system.

2.3.4 Course Structuring Model

A course is the set of educational activities chosen to represent a specific material to meet a very specific purpose. In the ALEM model, the structure of a course is modeled as an “and/or” tree, as it is shown in Fig. 2.5.

The course structuring model is composed of four types of nodes:

- Purpose: This is the final objective that a learner must reach at the end of the course. For example: revising UML within 10 days. A purpose is decomposed into several goals,
- Goal: This is part of the "and/or" tree that defines the intermediate goals between the purpose and activities. A goal can be decomposed into other goals and is realized by one or more activities,
- Activity: This is an operational goal. It defines a task that the learner must perform, such as acquiring a concept, solving a problem, listening to an audio clip. It must be performed to meet a goal. An activity is connected with components defined in the domain model,
- Component: This is the element on which an activity is executed. It represents the educational resources. It can be a concept, a page or a fragment.

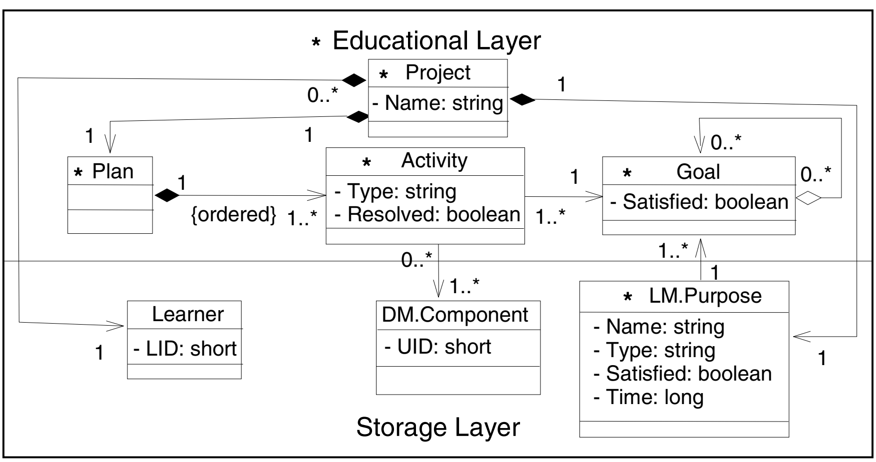


Fig. 2.5 Course structuring model

The course structuring model can represent the course structuring model. From a course “and/or” tree, the system generates a sequence of activities that the learner must follow to achieve the purpose. This sequence is called a plan (learning curriculum).

In this model, we define an educational project as a package that contains a purpose and a plan and which is assigned to a specific learner. A learner may have several projects in progress.

2.3.5 The Process of Generating the Appropriate Curriculum

After that the learner chooses a purpose, the system must perform the following steps to generate the appropriate learning curriculum:

- Update the model of the learner with the value chosen for the purpose,
- Select a subnetwork from the domain model that satisfies this purpose,
- Eliminate from the resulting network the components that are already acquired by the learner,
- Build the tree course model:
  - Position the components (concepts, page and fragments) in the bottom of the tree (leaves of the tree),
  - Set their strictly higher goals in level n-1 of the tree and name them as activities,
  - Add in the lowest levels of the tree (level < n-1) the higher level goals,
  - Add in level one (root of the tree) the purpose.
- Choose a path among the different possible paths to reach the purpose based on adaptation rules,
- Update the learner model by the values of project, activities, goals and plan.

2.4 UML Tutor System

To validate our reference model described previously, we defined an architecture and developed a prototype called UML Tutor which is used to teach UML. The prototype that we present in this section aims to propose an adaptive learning environment respecting the different concepts and processes of the ALEM reference model.

The prototype allows the possibility to integrate existing learning resources (documents) in the educational system. These resources are described using meta data respecting the LOM formalism. The system allows creating concepts, to annotate them and to link them to documents. The goal is to add semantic information to manipulate these resources to adapt them to the demands of the learner. The various entities of hypermedia are represented in a visual manner using graphs. We remember that all these information are stored in the domain model, the learner model, the adaptation model and the course structuring model.

In what follows, we will first define the different actors of the system and actions performed by these actors. We will then explain the application architecture. Finally, we will present the various modules of our prototype (UML Tutor) with the application GUI (Graphical User Interface).

2.4.1 UML Tutor Architecture

According to the ALEM reference model, there are two types of actors (system users): Teachers providing functionality for managing learning resources and learners who use the system to self train. To enable interoperability between UML Tutor system and other educational adaptive hypermedia systems we added a secondary actor named "other system".

UML Tutor system provides different application modules for that the three types of actors can perform their actions. Each module is related to a type of action. The system allows the learner to initialize his or her profile and perform self-training. It also allows the teacher to create and manage the hypermedia domain, courses structures and adaptation and acquisition rules. Application modules are grouped by type of user. Thus, the architecture consists of two applications offered to users. One used by the teacher to build adaptive hypermedia and another used by the learner to perform learning activities. Both applications are web applications made using the PHP language. UML Tutor contains a third application that is destined for other systems. It is the interoperability interface. It opens access to the models base to enrich it and to extract information. This application is made with web services.

These three applications are front end modules. They invoke internal modules that can perform basic functions for manipulating information stored in the four models of UML Tutor. The internal application modules are offered as web services. This will enable the modularity and ease of interoperability of the system.

For reasons ease of use of the application, we preferred to split the update of the domain model into two stages:

1. The construction and update of the objects "Concept", "Page" and "Fragment" via the Domain Editor,
2. The construction and update of the hierarchy of the domain goals (building adaptive courses) through the Course Editor.

Fig. 2.6 illustrates the software architecture of UML Tutor system showing the various modules of UML Tutor and interactions between them. This architecture also shows the two databases that the system uses which are the educational documents base and the models base.

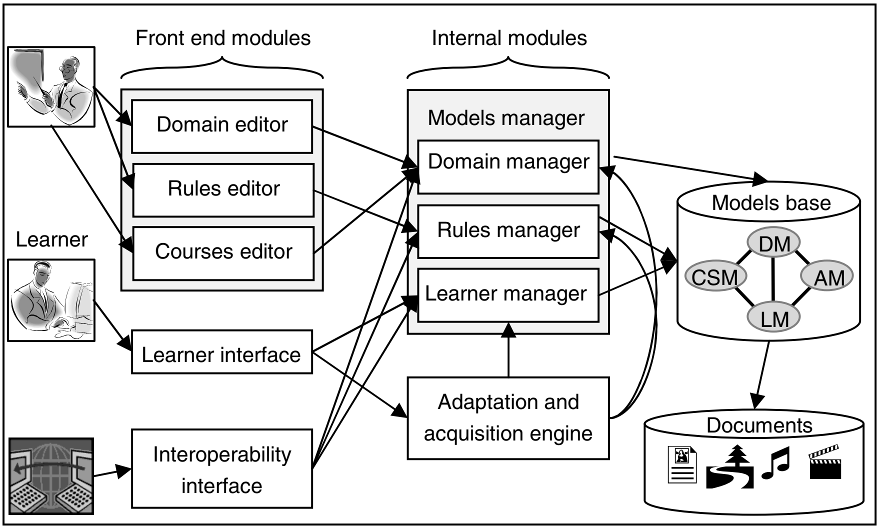


Fig. 2.6 UML Tutor architecture

2.4.2 Learning Process in the UML Tutor system

The following steps explain the different tasks performed by the learning environment, the learner and the teacher:

1. The teacher creates the domain model by filling the various concepts, pages, fragments and relations between them,
2. Creation of an adaptive course plan by the teacher describing a course that satisfies a goal (and/or tree),
3. The learner initiates his model by filling a quiz that focuses on his characteristics,
4. The learner selects a purpose,
5. The system generates a course plan adapted to the characteristics and the selected learner purpose,
6. The UML Tutor system presents to the learner the activities in the order of the adapted courses.

2.4.3 Modules of UML Tutor

In this subsection, we explain the role of various modules of the architecture of UML Tutor.

2.4.3.1 Domain Editor

This module allows teachers to manage the domain model via an intuitive web GUI (Fig. 2.7). This GUI allows building a graph composed of concepts, pages, fragments and relations between them. The composition of the model is done by drag and drop from the "Object" palette.

Each object in the palette is defined using several properties as follows: Concept (color, name and LOM description); Page (color, name, URL and LOM description); Fragment (color, name, URL and LOM description); Relationship (color and type). The description of these properties is presented as follows:

- Color: The display color of objects. It is used to make the model more readable,
- Name: The name of the object instance. It is unique throughout the system,
- URL: The path to the document which may be a local file (ex: "c:\documents...") or a remote file (ex: "http://www.UMLTu..."). The system offers the possibility to browse files on local and network directories,
- LOM Description: An XML clause which describes the object using the LOM formalism. The click on the button of this property displays a window which allows to describe the object using the 45 attributes of LOM grouped in 9 categories then transform them into XML format,
- Relationship type: The type of links that connect the various objects (composition, prerequisite, define, include, etc.).

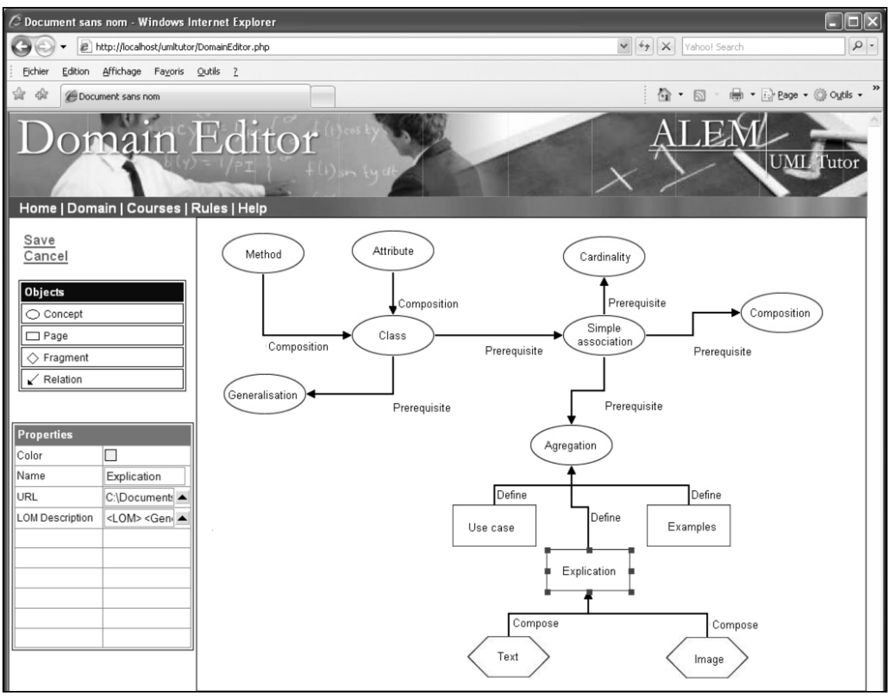


Fig. 2.7 Domain Editor

2.4.3.2 Courses Editor

The application module "Course Editor" allows creating and maintaining adaptive courses. As a reminder, this module does not interact with the structuring course model but with the domain model through the Domain Manager. The domain model consists of a forest of and/or trees (several adaptive courses). In this forest there are trees that can share the same nodes, which means that there may be activities, goals or resources that are common to several adaptive courses.

From the first page of this module the teacher can view the list of existing adaptive courses which are identified by their names and can also add, delete or modify an adaptive course.

Fig. 2.8 shows the detail page of an adaptive course represented in an “and/or” tree who is called "Class diagram". From this page it is possible to construct the tree using purpose, goal and activity objects by "drag and drop" items present in the objects palette. The purposes, goals and activities are transformed into goals when stored on the domain model.

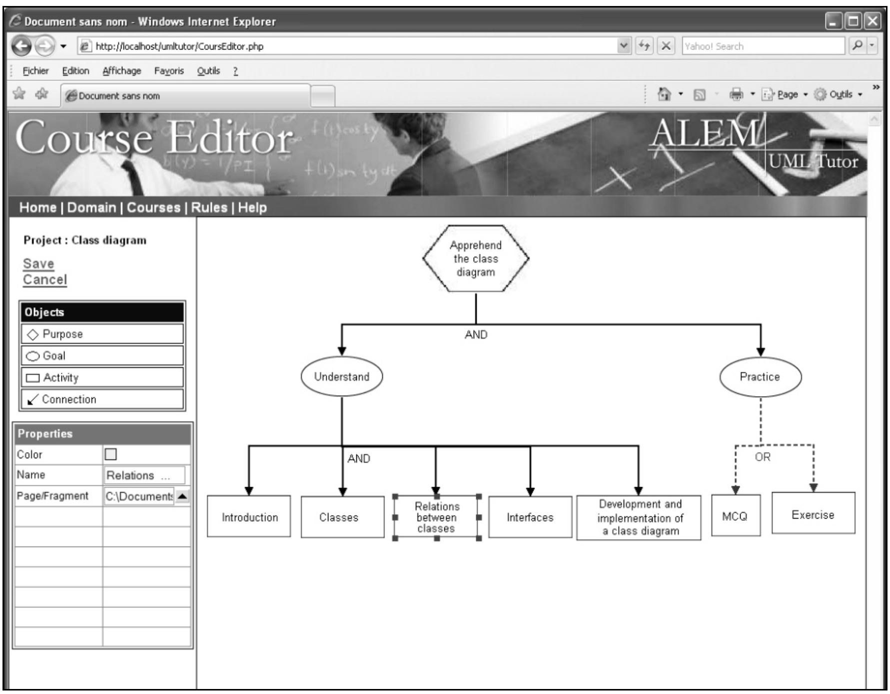


Fig. 2.8 Courses Editor

Each object is defined using several properties as follows: Goal (color and name); Activity (color, name and page/fragment); Relationship (color and type). The description of these properties is given below:

- Color: The display color of objects. It can be used to make the model more readable,
- Name: The name of the object instance. It must be unique throughout the system,
- Page/Fragment: The path to a page or a fragment which already exists in the domain model. To ease the teacher to identify the right page or fragment, the system offers the ability to search using the LOM attributes,
- Relationship type: The type of links that connect the various objects (AND, OR).

2.4.3.3 Rules Editor

This module is used by teachers to define the adaptation and acquisition rules. It interacts with the rule manager to maintain the adaptation model.

From the first web page of this module the teachers can view a list of available rules which are identified by their names and can also add, delete or modify a rule. Fig. 2.9 shows the detail page of a rule.

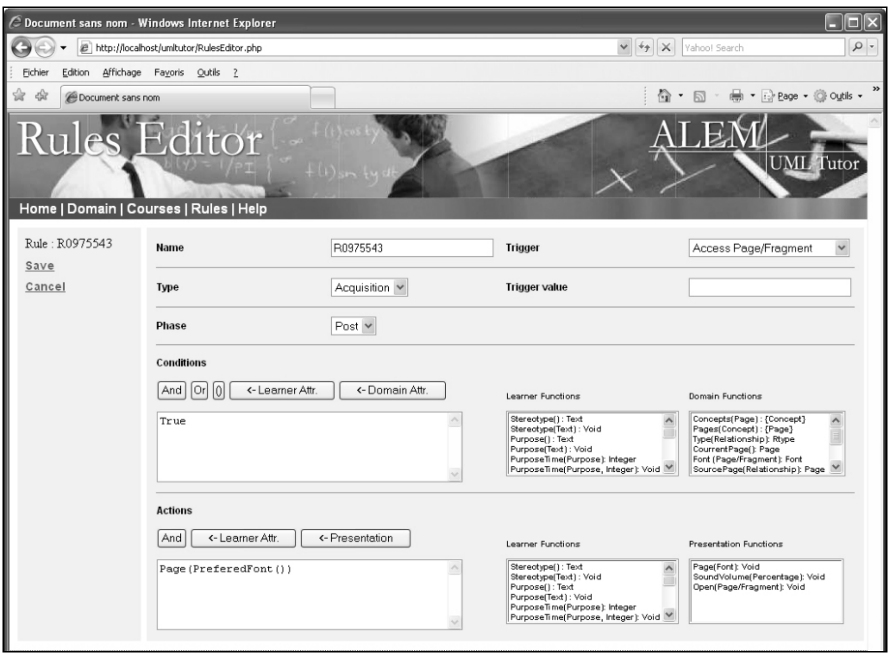


Fig. 2.9 *Rules Editor*

From this page it is possible to describe the properties of the rule, its conditions and actions:

- Name: The name of the object instance. It must be unique throughout the system,
- Type: The type of the rule, if it concerns an adaptation or acquisition rule. We remind that the first type is used to adapt the hypermedia to the requirements of the learner and the second is to update the learner model,
- Phase: The moment of the rule launching, before (Pre) or after (Post) the generation of the page and its display to the learner,
- Trigger: The event that triggers the rule. The UML Tutor system contains 5 possible triggers:
  – Access page/fragment: The rule is triggered when the learner accesses a page or a fragment,
  – Learner inactivity: The rule is triggered when the learner is idle for a specified duration. This duration is specified in the property "Trigger value",
  – Scroll page: The rule is triggered when the learner scrolls a page,
  – Learner input: The rule is triggered when the learner fill an input value. This value is specified in the property "Trigger value",
  – Executed rule: The rule is triggered after the execution of the rule specified in the property "Trigger value".
- Trigger value: The parameter for some types of triggers,
- Conditions: All the conditions for triggering the rule are separated by operators "and/or" and parentheses. The conditions can be domain functions or learner functions,
- Actions: The set of actions to be executed when a rule is triggered. These actions are separated by the operator "and" and might be either learner functions or presentation functions.

UML Tutor system provides to the teacher several functions for writing the conditions and actions of rules. The system can compose between different attributes and functions. Below are some attributes and functions grouped by type:
Domain functions: It holds the next attributes and functions:

- Concepts (Page): {Concept}, this function returns all the concepts related to the page specified as a parameter,
- Pages (Concept): {Page}, this function returns all the pages linked to the concept specified as a parameter,
- Relations (Concept/Page): {Relationship}, this function returns all the relationships related to the page or concept specified as a parameter,
- LOM (Concept/Page/Fragment, attribute): Text, this function returns the value of the specified LOM attribute about a concept, a page or a fragment,
- Font (Page/Fragment): Font, this function returns the font of the page or fragment specified as a parameter,
- SourcePage(Relationship): Page, this function returns the source page of the relationship specified as a parameter,

- DestinationPage(Relationship): Page, this function returns the destination page of the relation specified as a parameter,
- The above two functions also exist for the concepts: SourceConcept (Relationship): Concept and DestinationConcept(Relationship): Concept,
- Type(Relationship): Rtype, this function returns the type of the relationship specified as a parameter. The type can be prerequisite, variant, compose, etc.,
- CurrentPage(): Page, it returns the current page that the learner is browsing.

Attributes/characteristics of the learner: It contains the next attributes and functions:

- Stereotype(): Text and Stereotype(Text): Void, the first function returns the name of the stereotype of the current learner and the second updates it,
- Purpose(): Text and Purpose(Text): Void, the first function returns the name of the purpose that the learner has chosen and the second updates it,
- PurposeTime(Purpose): Integer and PurposeTime(Purpose, Integer): Void, the first function returns the time chosen by the learner to achieve the purpose specified as a parameter and the second update it,
- ReadPage(Page/Fragment): Boolean and ReadPage(Page/Fragment, Boolean): Void, the first function returns "True" if the learner have read the page or the fragment specified as a parameter otherwise the function returns "False" and the second updates the status of reading the page or fragment,
- ConceptAcquired(Concept): Boolean and ConceptAcquired(Concept, Boolean): Void, the first function returns "True" if the learner has acquired the concept specified by the parameter otherwise the function returns "False" and the second updates the status of acquisition of the concept,
- PreferedSoundVolume(): Percentage and PreferedSoundVolume(Percentage): Empty, the first function returns the sound volume preferred by the learner and the second updates it,
- PreferedFont(): Font and PreferedFont(Font): Void, the first function returns the value of the font preferred by the learner and the second updates it,
- CurrentActivity(): Activity and CurrentActivity(Activity): Void, the first function returns the current activity that the learner is achieving and the second set the activity specified as parameter as the next activity to execute,
- Page(Activity): Page, returns the page related to the activity specified as parameter.

Functions of the presentation: It embraces attributes and functions, such as:

- Page(Font): Void, this function is used to change the font of the page displayed to the learner,
- SoundVolume(Percentage): Void, this function changes the sound volume of the media that appears to the learner whether audio or video,
- Open(Page/Fragment): Void, this function opens the page set as a parameter.

2.4.3.4 Learner Interface

Learning interface is used by the learner to perform the necessary activities to achieve educational purposes. This module interacts with the "Learner manager" and the "adaptation and acquisition engine". The learner interface is divided into two parts (pages) as follows.

Edition of the learner profile: From this page the learner can set his profile (Fig. 2.10). The learner needs to update the greatest possible number of parameters for a better adaptation. Below are the 6 types of parameters of the learner profile in UML Tutor:

- Personal information: This page lists the general characteristics of the learner to set such as the first name, last name, school level and language,
- Knowledge of UML: This page lets to fill the level of understanding of domain dependent knowledge (UML),
- Knowledge on the Entity Relationship model: This is the page where the learner can fill the level of understanding of domain independent knowledge which can be useful for the adaptation,
- Presentation Preference: This page offers the ability to specify physical preferences of the learner such as the sound volume or the font,
- Cognitive Preference: It is the page where the learner defines his cognitive characteristics such as density of content, the preferred type of resources,
- Purpose/Project: On this page the student chooses one purpose from the goals of the domain model. UML Tutor offers multiple ongoing learning projects. Therefore, it is possible to switch from one project to another via a setting on this page.

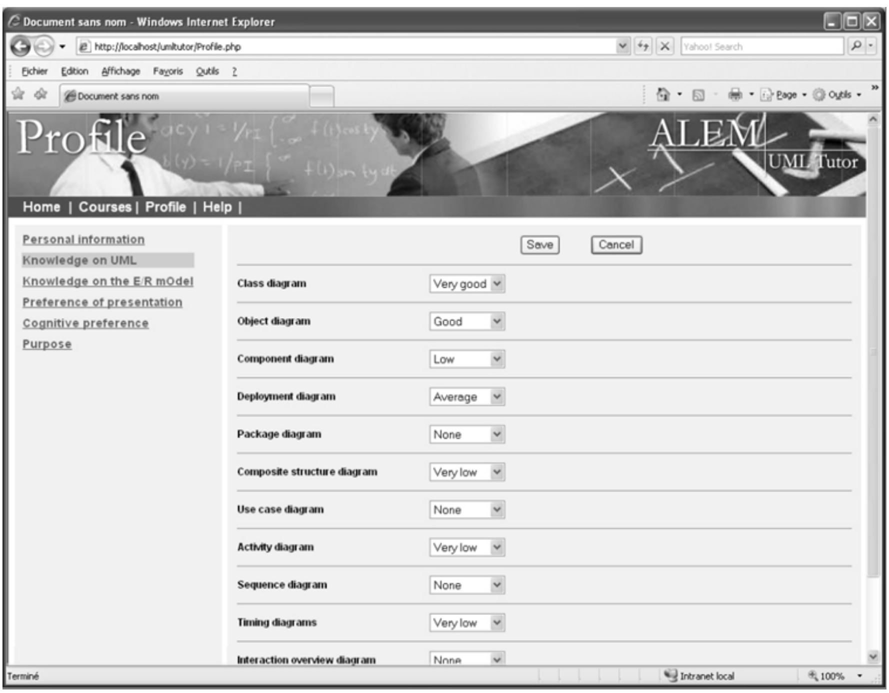


Fig. 2.10 Edition of the learner profile

Learning GUI: Once the learner chooses a purpose, the system generates a course plan adapted to his characteristics and his requirements. This course plan is displayed on the left page of the learning GUI (Fig. 2.11). Each line represents an activity such as the resolution of an exercise integrated into a web page, conducting a simulation, etc. The right frame shows pages related to the activities.

2.4.3.5 Interoperability Interface

This module is used by external systems to import or export the information stored in the four models of UML Tutor, i.e., the domain model, the learner model, the adaptation model and the course structuring model. To increase the accessibility of this interface and make it easily usable by external systems, we chose to develop it as a web service.

Fig. 2.11 Learning GUI

The interoperability interface conducts its extraction operations or updates via the domain manager, the rules manager and the learner manager.

2.4.3.6 Models Managers

This is a set of managers which are used by front-end modules and the adaptation and acquisition engine to perform basic operations of extraction or data update. These operations are executed by performing SQL queries (Insert, Update, Delete and Select) on the models base.

These managers are used to make transparent access to data and resources from other modules by providing a web service based interface. The three managers are explained below:

Domain Manager: The domain manager is a module that is used to manipulate the information on the domain model by performing some basic operations. It is used by three front-end modules that are the domain editor, the course editor and the interoperability interface and an internal module which is the adaptation and acquisition engine. The basic operations of this module can be: add a concept, a page or a fragment; add a goal and assign it to one page; delete a relationship between a concept and a page; describe a fragment by using a LOM descriptor; etc.

Rules manager: The rules manager is a module that is used to manipulate the information present on the adaptation model by performing some basic operations. It is used by two front end modules that are the rules editor and the interoperability interface and an internal module which is the adaptation and acquisition engine. The basic operations of this module can be: add a rule; change a property of a rule; extract the conditions of a rule; extract all the rules that have a specific type of trigger; etc.

Learner manager: The learner manager is a module that is used to manipulate the information on the learner model and the structuring course model by performing some basic operations. It is used by two front end modules that are the learner interface and the interoperability interface and an internal module which is the adaptation and acquisition engine. The basic operations of this module can be: add a purpose; update the preferred font of the learner; update the level of knowledge of a concept; extract the sequence of activities that the learner must perform for a given purpose; etc.

2.4.3.7 Adaptation and Acquisition Engine

The role of the adaptation and acquisition engine is to execute the rules present on the adaptation model. To perform this task, this module uses the information stored in the four models of the base and that via the four managers. The engine performs the following four steps to make the necessary adaptation or acquisition:

1. Stay tuned to capture any event that may trigger a rule. As explained earlier, these events can be an access to a page, scrolling a page, a keyboard input, etc.,
2. Identify the rules that use the type of the detected trigger,
3. Check for the identified rules if the conditions part is verified,
4. If so the engine must perform actions otherwise do nothing. If this is an adaptation rule, the engine must present the appropriate adapted resource to the learner and if it is an acquisition rule, the engine must update the necessary information in the models base.

2.4.3.8 Models Base

The models base is the repository where is stored the data related to the domain model, learner model, structuring course model and adaptation model. This base can be constructed via Extended Markup Language (XML) files or via a relational database. In our application we chose to use the second solution.

In the UML Tutor system, the relational data model is based on the UML diagrams of the four models defined on the ALEM reference model. Then each of the four UML class models of ALEM is represented by a table.

2.4.3.9 Learning Resources Base

The learning resources base represents a container that includes all the files used by UML Tutor. These resources can be of different natures: web Page, image, PDF Document, Flash application, etc. and may be available either on the local network or on the web.

When a teacher updates the domain model and specifies a resource stored locally on his own machine, the system uploads it on the resource base. Therefore, it will be shared between teachers and learners. When the teacher uses network or web resources, they are not downloaded on the resources base but they are considered as resources that belong virtually to the resources base.

2.5 Case Study

Education reform is one of the largest workshops in e-Algeria program (2009-2013). Indeed several Algerian universities already have a platform for distance education such as “UABT-EAD", a Moodle platform, of the UABT University (http://www.univ-tlemcen.dz). They aim to improve the quality of learning of their students within the new educational system called “LMD: License-Master-Doctorate”. The LMD system requires a lot of personal and cooperative work through learning management system platforms and other e-learning tools.

Thus, the UML Tutor application presented in Sect. 2.4 is being used in a software engineering course for level 2 master students at the Department of Computer Science of the University of Tlemcen. The number of students taking this course is about twenty students, with very heterogeneous profiles. For example when the majority of students are pure LMD students having a LMD license, some students were converted from the old yearly educational system having an engineer degree. There are also some part time students coming from industry. Also, the tracks followed by each student are not the same.

The authors conducted a small questionnaire with active course students. Their aim was to know how the students were using the UML Tutor application. One of the members stated that: "By using the tool I was able to gain insight over the most interesting aspects of the course. More than that, I was then able to personalize my learning in my own context". Another student found that the tool gives him the right resources he needs and it skipped all the concepts that he knows. Some students found that the fact to specify the time to achieve a goal before starting the course is a very interesting feature. On the side of teachers, some ones have found that the domain editor and the course editor are very intuitive tools that helped them to structure their teaching resources.

This qualitative analysis shows that the UML Tutor application would be very well accepted in the future by the students as well as by the instructors.

2.6 Conclusions

Classic hypermedia systems provide the same pages to different users even if the user population has different knowledge, different goals, etc. These users need pages that are adapted to their profiles and their requirements. The Adaptive Hypermedia Systems are trying to overcome the problems of the classic hypermedia using the knowledge present in the learner model to adapt the content and links that are presented to a learner.

Existing reference models for modeling AHS are generic and are not dedicated to educational systems. In this paper, we introduced a reference model specific for educational AHS that we named ALEM (Adaptive Learning Environment Model). The great contribution of our model is modeling the personalized course plan of the learner. The architecture of our model is similar to reference models AHAM, Munich and Social LAOS.

In ALEM, we use four models to adapt the hypermedia to the needs of the learner: 1) the domain model which is a representation of the subject of hypermedia by concepts and links between them; 2) the learner model that represents user characteristics and needs; 3) the adaptation model which contains rules for adaptation; 4) the structuring course model that specifies the set of activities that meet one or more educational goals.

To validate our reference model we have presented an educational AHS for teaching UML called UML Tutor which is an application of our reference model. The four models of the ALEM reference model were defined in this system.

The prospects of this work are to finalize the ALEM model and improve it. Among possible improvements, the following are the most important:

- Make the ALEM model less specific and more extensible,
- Link the ALEM model with specifications of the consortium IMS Global IMS-Learning design (IMS 2003) and learner information packaging (IMS 2001) for a better semantic interoperability,
- Define basic adaptation rules that can be used by any educational adaptive hypermedia system,
- Describe the rules engine in detail and the process used for the execution of these rules.

References

Brusilovsky, P., Millán, E.: User Models for Adaptive Hypermedia and Adaptive Educational Systems. In: Brusilovsky, P., Kobsa, A., Nejdl, W. (eds.) Adaptive Web 2007. LNCS, vol. 4321, pp. 3–53. Springer, Heidelberg (2007)

Campbell, B., Goodman, J.: HAM: A general purpose hypertext abstract machine. Communications of the ACM 31(7), 856–861 (1988) de Bra, P.: Adaptive hypermedia. In: Adelsberger, H.H., Kinshuk, P., Pawlowski, J.M., Sampson, D. (eds.) Handbook on Information Technologies for Education and Training, 2nd edn., pp. 29–46. Springer, Heidelberg (2008)

Ghali, F., Cristea, A.I.: Social Reference Model for Adaptive Web Learning. In: Spaniol, M., Li, Q., Klamma, R., Lau, R.W.H. (eds.) ICWL 2009. LNCS, vol. 5686, pp. 162–171. Springer, Heidelberg (2009)

Halasz, F., Schwartz, M.: The Dexter hypertext reference model. Communications of the ACM 37(2), 30–39 (1994)

IEEE Learning Technology Standard Committee. Learning object metadata (LOM) (2005), `http://ltsc.ieee.org/wg12/` (accessed November 01, 2010)

IMS Global Learning Consortium. Learner information packaging (LIP) (2001), `http://www.imsglobal.org/profiles/lipinfo01.html` (accessed November 01, 2010)

IMS Global Learning consortium. Learning design (LD) (2003), `http://www.imsglobal.org/learningdesign/ldv1p0/imsld_infov1p0.html` (accessed November 01, 2010)

Knutov, E., de Bra, P., Pechenizkiy, M.: AH 12 years later: a comprehensive survey of adaptive hypermedia methods and technique. New Review of Hypermedia and Multimedia 15(1), 5–38 (2009)

Koch, N., Wirsing, M.: The Munich Reference Model for Adaptive Hypermedia Applications. In: Proceedings of ICAHAWBS (2002)

Object Management Group. UML: The Unified Modeling Langage (2005), `http://www.omg.org/uml` (accessed November 01, 2010)

Papanikolaou, K.A., Grigoriadou, M., Kornilakis, H., Magoulas, G.D.: INSPIRE: An INtelligent System for Personalized Instruction in a Remote Environment. In: Reich, S., Tzagarakis, M.M., De Bra, P.M.E. (eds.) AH-WS 2001, SC 2001, and OHS 2001. LNCS, vol. 2266, pp. 215–225. Springer, Heidelberg (2002)

Tadlaoui, M., Chikh, A., Bouamrane, K.: ALEM: Un modèle de référence pour les applications web adaptatif educatif. In: Proceedings of e-LfaC (2010)

Wu, H.: A reference architecture for adaptive hypermedia applications. PhD dissertation, Eindhoven University of Technology (2002)

Abbreviations

AHS	Adaptive Hypermedia Systems
ALEM	Adaptive Learning Environment Model
AM	Adaptation Model
CSM	Course Structuring Model
DM	Domain Model
GUI	Graphical User Interface
LID	Learner IDentifier
LM	Learner Model
LOM	Learning Object Metadata
SQL	Structured Query Language
UML	Unified Modeling Language
URL	Uniform Resource Locator
XML	Extended Markup Language

Chapter 3
Proactive Sequencing Based on a Causal and Fuzzy Student Model

Alejandro Peña-Ayala[1,2,3] and Humberto Sossa[4]

[1] WOLNM, 31 Julio 1859, No. 1099-B, Leyes Reforma, Delegación Iztapalapa, 09310, DF, Mexico
apenaa@wolnm.org
[2] Escuela Superior de Ingeniería Mecánica y Eléctrica, Unidad Zacatenco, IPN Av. Miguel Othon de Mendizabal, S / N, Escalera, Gustavo A. Madero, 07738, DF, México
apenaa@ipn.mx
[3] Osaka University, Institute of Scientific and Industrial Research, 8-1 Mihogaoka, Ibaraki, Osaka, 567-0047 Japan
apenaa@ei.sanken.osaka-u.ac.jp
[4] Centro de Investigación en Computación-IPN, Av. Juan de Dios Bátiz s/n, México, DF, México
hsossa@cic.ipn.mx

Abstract. Proactive education paradigm pursues to infer future possible events and states of the teaching-learning cycle to accomplish better students' apprenticeship, and overcome likely issues. An essential functionality to implement such a paradigm is the prediction. Thus, our approach aims at anticipating student's domain knowledge (DK) acquisition through the development, and use of a causal and fuzzy student model (CFSM). The CFSM depicts several domains of student's attributes, that are taken into account for sequencing lectures to students. Moreover, it also characterizes attributes of the content, to shape the nature of the available lectures authored to teach a given concept. Both sorts of attributes are defined semantically as concepts in an ontology. These concepts set causal relationships between each other. This type of relationships represents a belief of how an attribute exerts the status and activation of another attribute. Concepts and causal relationships are sketched as a cognitive map (CM). The description of the attributes and the causal relationships are respectively made by fuzzy values and fuzzy rules-bases. Linguistic terms instantiate the state of concepts and a version of fuzzy-causal inference is fulfilled to produce causal behavior and outcomes about the state of the concepts. Based on these elements, our approach simulates the learning results that a lecture could produce on student's apprenticeship. Such a prediction is accounted to choose the most profitable lecture for being delivered to student. As a result of an experiment, we found out those users of a web-based

A. Peña-Ayala (Ed.): Intelligent and Adaptive ELS, SIST 17, pp. 49–76.
springerlink.com

educational system (WBES) that sequences lectures based on the advice given by the CFSM reached 17% higher learning than their peers who did not have the support of our approach. So in this work, we highlight the attributes of the approach.

3.1 Introduction

A proactive functionality in WBES claims: Decisions concerning to the authoring, selection and delivery of a wide sorts of items and events involved in a teaching-learning experience (e.g., kind of content, type of navigation, interactivity protocol, trial to develop, quiz to request, evaluation to apply, assessment to be made…) are made based on the anticipation of students' reactions, behavior, and outcomes (Sims et al. 2002, Duke and Madsen 1991, Segravea and Holta 2003).

Such a principle is accounted by Zampunieris 2006, 2008, who proposed a proactive learning management system (LMS) based on a dynamic rules-based system. It helps users to better interact online by providing programmable, automatic, and continuous analyses of the users' actions, augmented with appropriate tasks initiated by the LMS itself. An enhanced version of his approach includes an algorithm, to make lazy evaluation, to avoid unnecessary and time-costly requests to the LMS database when a rule is not activated.

According to Merceron and Jacef 2005, data mining (DM) algorithms can be used to discover pedagogically relevant knowledge contained in databases, in order to implement proactive functionality in WBES. They assert the findings are useful for helping teachers and WBES to manage their class, understand their students' learning, and reflect on their teaching. Likewise, DM supports learner reflection and provides proactive feedback to learners.

Concerning sequencing, it is stated by the Naval Education and Training Command (NAVEDTRA) as: The arrangement of learning objectives for a course into a logical teaching sequence (NAVEDTRA 1997). Moreover, in the Learning Technology Systems Architecture (LTSA), proposed by the Learning Technology Standards Committee (LTSC) as a working group of the Institute of Electrical and Electronic Engineers (IEEE), sequencing is a subject area carried out by the system coach component of the LTSA. It interacts with others LTSA components as follows: accounts the learning preferences of the learner entity, examines the assessments outcome by the evaluation, and commands the delivery to provide the suitable learning content to student (IEEE-LTSA 1999).

One interesting sequencing model is the First Principles of Instructional Design designed by Merrill 2002. It links a taxonomy of learning object (LO) types, which provides support for making scoping decisions, to a sequencing structure. In the Merrill's model LO types are defined by instructional function. It advices to author five types of LO to be delivered in the next sequence: 1) presenting the real-world problem the student will be engaging; 2) triggering prior DK; 3) acquiring new DK; 4) practicing with feedback; 5) promoting transfer and integration.

In addition, Zapata-Rios 2006 designed a useful process to sequence a set of LO, based on a content analysis technique, which follows the next three steps: 1) discover and highlight the main axis of the content student should learn: 2) discover and highlight the main contents and organize them in a hierarchical and relational structure; 3) sequence contents according to the principles of the psychological organization of knowledge.

As regards with student model (SM), John Self (1996) asserts: A SM is a representation that an intelligent tutoring system (ITS) has of the student using the system. Likewise, student modeling is a special process of user modeling which is relevant to the adaptability of ITS (Elsom-Cook 1993).

However, a SM is also a part of any WBES, LMS and other kinds of educational and hypermedia systems, which are adaptive to meet students' needs. The SM is acknowledged as the learner model subject area of the LTSA. It is represented by the learner entity LTSA component and collaborates with other LTSA components in the next way: student's behavior is monitored by the learner entity and the evaluation, student's learning preferences are elicited by the learner entity and taken into account by the system coach, and student is the object of interaction with the multimedia content provided by the delivery (IEEE-LTSA 1999).

Self 1994 identifies several key uses of the SM, such as the next: 1) prediction: SM need to be predicted by the reasoning and learning items of the SM, so items of the current SM map the SM and the sequence of events onto a new SM: 2) evaluation: the states which the SM predicts that could be reached need to be evaluated, so the evaluation is determined by a function of SM with respect to prespecified objectives or intrinsic properties of the SM; 3) planning: the educational system must be capable of dynamic planning, that is, the on-line creation and revision of instructional plans, since for any significant learning context the prespecification of a plan to be strictly followed is not desired.

A sample of predictive SM is the approach developed by Timms 2001. It uses information about student ability based on pre-test data to predict when a student is likely to need help in advance of the student requests it. Moreover, an instance of fuzzy SM is the neuro-fuzzy system built by Sevarac 2006. It enables classification of students based on qualitative observations of their characteristics. In addition, a causal SM is stated by Millan et al. 2001. It uses Bayesian networks and causal independence to factor the conditional probabilities and decreases the parameters for each question to a number linear in the amount of concepts.

With the aim at intelligently supporting a WBES behaves adaptive, our approach pursues to implement proactive sequencing based on a CFSM. It deploys proactive functionality by the anticipation of students' learning achievements based on predictions outcome by fuzzy-causal reasoning. The knowledge base of the WBES can be mined by DM algorithms in order to explain and anticipate behaviors. The sequencing is made by the prediction of the bias that a LO could exert on the student's learning, the evaluation of the available LO to teach a given concept, and the selection of the one that offers the highest student's apprenticeship. Instead of using neuro-fuzzy and probabilistic reasoning our approach applies a fuzzy-causal reasoning engine for dealing with qualitative knowledge.

In order to describe our approach, the remaining of the chapter is organized as follows: The framework is outlined in section 2 and underlying elements are given in section 3. Knowledge acquisition and representation are respectively set in sections 4 and 5. Cognitive mapping is detailed in section 6 and the simulation is depicted in section 7. An experiment is presented in section 8, whereas the conclusions and the future work are stated in section 9.

3.2 Conceptual Model

Our approach embraces the sequencing and the learner model subject areas of the LTSA. It implements the processes defined for the system coach and the learner entity LTSA components. Thus, the approach is designed as a Multi-Agent System (MAS), where agents perform specific services, manage knowledge repositories, and communicate each other by messages (Wooldridge, 1992). So an overview of the services and a profile of the main algorithm are given in this section.

3.2.1 Architecture

Our approach is an open system that provides sequencing and user modeling support to educational systems, such as: WBES, LMS, and ITS. It holds independent modules that are deployed as web services. They are invoked by messages whose content is encoded by means of the Intelligent Physical Agents (FIPA) Agent Communication Language (FIPA, 2001). The architecture is organized as follows:

- Acquisition of student's profile: It embraces specialized agents to elicit, evaluate and depict knowledge of a specific domain (e.g., cognitive skills, personality traits, DK…) to characterize a student. The acquired domains shape a student's profile. Such a profile is stored as a knowledge repository of the SM.
- Acquisition of content's profile: It provides an agent for helping user to qualify attributes of the content from several viewpoints, like the type of LO and linguistic level. Based on these criteria, the content's profile of the DK is tailored. The profile is organized as a knowledge repository to describe content.
- Semantic management: It devotes an ontology-agent to acquire, maintain and access the meaning of concepts and causal relationships. It administrates an ontology encoded through the Web Ontology Language (OWL). The semantic repository owns the statement of meta-classes, classes, attributes, and instances.
- Causal relationships setting: It has an agent aimed at assisting users to set a topology of causal relationships. Attributes of the content's profile are *cause concepts* that stimulate or inhibit attributes of the student's profile, known as *effect concepts*. The concepts' state is instantiated by linguistic terms, named *universe of discourse* (UOD), and causal relationships are defined by *fuzzy-causal rules bases* (FCRB). Both UOD and FCRB are declared in an ontology.

- Cognitive mapping: It includes an agent able to interpret the student's and content's profiles, the data repositories and the ontology in order to sketch a CM. The topology of the CM is stored in a repository as a network of nodes and links to show how a node, a cause concept, biases another, an effect concept.
- Causal simulation: It contains agents to fulfill the tasks needed to induce fuzzy-causal conclusions. It uses a working-memory to recreate a simulation environment of causal effects, and a repository to store the values that represent how the concepts' state evolves through the time, as a result of the causal bias.
- Sequencer: It accounts an agent that chooses the series of lectures to be delivered to student. At each stage, it identifies the candidate LO, those authored to teach a concept. Next, it evaluates the results outcome by the fuzzy-causal simulation achieved on the CM tailored to each candidate LO. After, the agent chooses the LO that offers the highest student's learning accomplishment.

3.2.2 Algorithm

Based on the prior architecture and the sequencer component, in this section, the algorithm devoted to take over the teaching-learning cycle is shown. The main tasks to be carried out are stated at the end of this subSect.

In lines 01 to 02, the student's and content's profile repository are accessed, whereas the evaluation of LO authored to teach a concept is controlled in line 03. Repositories of causal relationships and semantic definitions are used for cognitive mapping in line 04. Lines 05 to 16 describes the tasks fulfilled by the fuzzy-causal engine. Simulation of a causal effects is made through discrete increments of time, lines 06 and 15, until a stable situation is reached, lines 07 and 13-14. A new value is estimated for each concept's state of the CM according the causal relationships that bias on it, lines 09-12. The concepts' state values are stored in line 16.

Once the simulation of all the instances of CM ends, it is chosen the CM whose outcomes offer the highest learning accomplishment, lines 18 to 21. It corresponds to the final value attached to the concept that represents the learning gained by the student for the current topic to be taught, line 20. Next the chosen LO is delivered.

```
00: Algorithm used for sequencing the best LO
01: Access student's profile of the student e
02: Access content's profile of the DK concept c to be taught
03: For each candidate LO lo authored to teach concept c do:
04:  Do the cognitive mapping to outcome an instance: CMe-lo
05:  Do the fuzzy-causal engine to simulate CMe-lo behavior:
06:   Set time = 1 and stability = false
07:   While (time < 100 and not (stability)) do:
08:    For each effect concept ce of the CMe-lo:
09:     For each causal relationship r that heads to ce do:
10:      Estimate the level or/and variation effect on ce
11:     Compute the new state of ce by fuzzy-causal inference
12:     Track behavior and final states' values of ce
13:    If (the current concepts' states are already stored)
14:     Set stability = true
```

```
     Set time++
   Track the causal behavior and outcomes of CMe-lo
 Set b = 1, the first CM as the current best option
 For each option CMi, since the second to the last one do:
   Compare the outcomes of CMb against CMi
   If (CMi offers higher learning than CMb) set b = i
 Deliver the LO b to teach the DK concept c to student e
```

3.3 Underlying Elements

Seven subjects shape the grounds of the approach. They offer a conceptual and a formal baseline to provide soundness. Thus, a cognitive process to define concepts and judgments is given. Next, the essentials of causality and CM are stated. Later on, a formal model for the SM is shaped and a profile of fuzzy knowledge is sketched. Finally, fuzzy and fuzzy causal reasoning are explained.

3.3.1 Generation of Concepts and Judgments

Essentially speaking, the SM is a mental representation of several sorts of attributes about a given learner. Usually, those attributes are set as concepts and judgments that respectively reveal properties of the individual and qualify personal traits. Thus, a cognitive process to outcome concepts and judgments is stated next.

The development of concepts is fulfilled along two stages: sensory and logic. Sensory stage is split into three cognitive processes: 1) sensation: the entities of the surrounding environment input stimuli to the individual through her/his senses, thereafter the stimuli are decomposed into parts; 2) perception: shapes the entity that gives out the stimuli based on its parts; 3) description: encodes a mental frame to characterize the entity that sends out the stimuli, afterwards it is stored and recalled without the presence of the prior stimuli (Miguelena 2000).

During the logic stage four processes are fulfilled: 1) analysis: splits the mental frame that represents the entity into components; 2) abstraction: highlights the key components of the entity; 3) synthesis: assembles the components in a mental scaffold of relevant attributes; 4) generalization: includes necessary attributes and removes irrelevant ones to tailor a common frame for the entities of a given class.

As a result, new concepts about entities of a domain are set up in the mind of the individual. Thus, a concept is a kind of thought that depicts main properties of entities and their internal relationships. Likewise, a *concrete* concept represents entities and classes of entities; whilst an *abstract* concept emphasizes properties and sketches relationships among entities such as causality and influence.

Regarding to relationships, people develop their consciousness by the interpretation given to the incoming stimulus and the establishment of beliefs about a particular domain. Such cognitive processes are admitted as judgments about properties and relations between entities. Thus, a judgment is a kind of thought to assert or deny an entity owns or lacks a property, or holds or misses a relationship.

The structure of the judgment is conformed by three elements: 1) subject: identifies the object of the judgment; 2) predicate: reveals what that it is said about the subject; 3) link affirms or rejects the ownership of the predicate by the subject. Two dual attributes are held by a judgment: affirmation-negation and true-false. A judgment can be singular, partial or universal; whether the thought is about only one entity, several entities, or all entities of a class respectively (Peña et al. 2008).

3.3.2 A Profile of Causality

Teaching-learning process can be represented as a cause-effect relationship, where a tutor or an educational system provides teaching stimuli to students with the aim at transferring DK and developing cognitive-motor skills. So causality is viewed as a cognitive phenomenon that happens in the course of a lecture. Thus, our approach takes into account causality according to the following baseline.

The philosophical principle of causality claims: Any fact has a cause, and given the same conditions, the same causes produce the same consequences (Carvalho 2001). The principle is externalized by causal relationships. They are stated as: A judgment to assert the existence of a tie between two concepts, where the *antecedent* concept exerts the *consequent* concept, accordingly its state is likely altered.

Concepts depict entities and events of the subject to be modeled. A tuple is stated to handle a concept. It holds a term to label the concept, a semantic definition, and a state to reveal a *level* of presence that the concept shows at a given time, or/and a *variation* of the change succeed on the concept after a while.

Causal relationships sketch the direction ($\_\rightarrow$) and the kind of bias (b) that a *cause* concept (c_c) exerts on an effect concept (c_e) (e.g., $c_c \;\_b_{c\text{-}e}\rightarrow c_e$). They also depict indirect relationships between concepts when appear intermediary concepts (c_i) in the causal link (e.g., $c_c \;\_b_{c\text{-}i}\rightarrow c_i \;\_b_{i\text{-}e}\rightarrow c_e$). Feedback is drawn when an effect concept also exerts a cause concept (e.g., $c_c \;\_b_{c\text{-}e}\rightarrow c_e \;\_b_{e\text{-}c}\rightarrow c_c$ or $c_c \;\_b_{c\text{-}i}\rightarrow c_i \;\_b_{i\text{-}e}\rightarrow c_e \;\_b_{e\text{-}c}\rightarrow c_c$). Self-feedback relationships are setting when a concept biases itself (e.g., $c_c \;\_b_{c\text{-}c}\rightarrow c_c$). The kind of bias is fully described by a FCRB.

Causal inference accounts an underlying assumption that claims: Given an event c_c at t_i, the event c_e at t_j can only be a consequence from c_c, if and only if $t_i < t_j$. Causal inferences alter the state of the concepts; but, they do not estimate the *absolute* state value of the concept neither consider its current state during the inference. When several cause concepts bias with the same sense to an effect concept, an *accumulative effect* is outcome. It produces a greater effect than a summative effect (e.g., if $b_{c\text{-}i}$ and $b_{i\text{-}e}$ represent *increases* and *increases much*, then the accumulative effect is *increases a lot*). Once the concepts' states are instantiated, inference starts along discrete increments of time to gradually transform the values until they reach a stable situation, a pattern of values or meet chaotic attractors.

3.3.3 Cognitive Maps

We propose CM as a tool to describe and simulate the teaching-learning cycle. Thus, our CFSM is sketched as a CM, where attributes of the student and content profiles are represented as nodes, the causal influence that those concepts exert each other are outlined as arcs, and the prediction of how a LO could bias the student's learning is outcome by the causal inference. The estimated behavior and results are accounted by the proactive sequencing to choose the LO to be delivered.

Essentially, a CM is a mental model of how an individual perceives a domain of analysis. The causal inference of a CM is able to model qualitative dynamic systems, as it involves the entities and events of the domain into a topology of cause-effect workflow. In addition, CM includes a causal inference engine, that simulates how the concepts bias themselves and their states evolving through time.

3.3.4 Formal Representation of the Student Model

A SM depicts beliefs held by the educational system about a learner. Beliefs are set by formulae in the propositional calculus. They reveal the behavior of an agent. According to Self 1994, the beliefs are formalized as follows:

Let p be a proposition where $B_s p$ holds: if system (s) beliefs (B) a proposition p. Thus, the set of beliefs about the user (U) that a system asserts is stated by: $B_s(U) = \{p \mid B_s p(U)\}$. But, beliefs are organized into domains. Thus, for every domain (D) that the system accounts to shape a student, it is set the notation: $D_s(U) = \{ p \mid B_s p(U) \cap p \in D\}$. Due to our SM embraces five domains, each of them uses the prior notation. The complete SM is the result of the union of five sets of propositions, such as: $SM_s(U) = \{L_s(U) \cup P_s(U) \cup C_s(U) \cup DK_s(U) \cup LO_s(U) \}$, where L, P, C respectively identify learning preferences, personality and cognition.

3.3.5 Fuzzy Knowledge

Based on the work made by Zadeh 1988, a profile of fuzzy logic is resumed in this subSect. As a first assumption, concepts are handled as linguistic variables. In this way, the state of the concepts can be instantiated by linguistic terms, called fuzzy terms. With the aim at making computations with words, they are mathematically outlined as fuzzy sets. Fuzzy sets are sketched as polygons to estimate a membership degree (MD) for a given value. In our approach, we chose the trapezoid and characterized it by the next attributes: shape, area, axis of central mass, support set, support set length, core set, core set length, interior base length, exterior base length, interior declination, and exterior declination (Peña et al. 2008).

As several fuzzy terms could instantiate a linguistic variable, they set an UOD. It is an array of fuzzy terms that are ordered according to their semantic sense of increasing level or variation. Thus, the UOD is also graphically drawn as a series of polygons on a Cartesian plane, where the abscissa (i.e., axis x) represents the degree of uncertainty and the ordinate (i.e., axis y) depicts the MD. In our approach, a level UOD holds a range of [0, 1] for x and y; whereas, in a variation UOD the range is [0, 1] for x and [-1, 1] for y. Moreover, the next *dimension criterion* is accounted: As much as a fuzzy term reveals a high level or variation, its corresponding area and the top set length must be longer, as well than the ones given for those fuzzy terms that revel smaller levels or variations (Carvalho 2001).

In order to instantiate some concepts of our SM, we set a *5-level* UOD with the fuzzy terms: {too low, low, normal, high, too high}. We also use an *11-variation* UOD to own fuzzy terms like {decreases: so much, much, moderate, little, so little, null, increases: so little, little, moderate, much, so much} (Peña 2008).

Fuzzy rules are split into an antecedent and a consequent. We adopt this structure to set fuzzy-causal rules, but constrain them to respectively, involve one cause concept and one effect concept. Moreover, the antecedent may involve level or variation fuzzy terms; but, the consequent only concerns to variation fuzzy terms. The antecedent is a condition to inquiry if the concept's state holds a specific fuzzy term at time t_i; whereas, the consequent is a conclusion to assign just one fuzzy term to the concept's state at time t_{i+1} when the rule fires (Carvalho 2001).

The whole causal relationship between a pair of concepts is described by a FCRB. It embraces just one rule for each fuzzy term of the UOD attached to the cause concept. When fuzzy-causal inference is fulfilled, at every increment of time t_i, the complete FCRB is scanned to identify, which rule(s), up to two, fire(s).

3.3.6 Fuzzy Reasoning

Some mechanisms designed to make fuzzy inference (Zadeh 1988) and the adaptations proposed by Carvalho 2001 are resumed in this subSect. However, we suggest reviewing the mathematical baseline and the graphical representation provided in those references and in Peña et al. 2008, 2012.

The first procedure corresponds to the generation of the *preliminary fuzzy set* (PFS). Once it is identified, the rule(s) that fire(s) at t_i; the fuzzy term(s) involved in the consequent(s) is (are) analyzed based on the next cases:

1. Just one rule fires: Then the PFS represents the fuzzy term stated in the rule's consequent and the dimension of its shape corresponds to the MD of 1.
2. Two rules fire and involve the same fuzzy term in their consequent: Then two PFS are set to depict the fuzzy term stated in each rule's consequent and the scale of their shape corresponds to their respective MD (i.e., less than 1.0).
3. Two rules fire and involve different fuzzy terms in their consequent: Then two neighboring PFS are outcome, each one expresses its respective fuzzy term involved in the consequent and its size is equivalent to its respective MD.

The second procedure is devoted to estimate the *outcome fuzzy set* (OFS). After the generation of the PFS, one of the next sceneries could succeed:

1. When there is just one PFS, then it produces an OFS with the same shape.
2. When there are two PFS and both concerns to the same fuzzy term, then an OFS shape alike to them but its dimension corresponds to the MD of 1.
3. When there are two PFS and both concerns to the neighboring fuzzy terms, then an OFS is set as a result of an *aggregation* process, but its shape is not similar to one of them, neither its dimension corresponds to the MD of 1.

The third procedure is oriented to compute a kind of *reinforcement effect*. This effect is only applied to fuzzy inference relationships (FIR). It estimates as the intensity of the MD is reinforced as a result of several FIR over a given concept (i.e., several cause concepts exert a fuzzy bias on a specific effect concept). Thus, it represents a kind of accumulation of the prior generated OFS on the ordinate. Based on the earlier three situations, the inferred fuzzy set (IFS) holds the same shape with the MD of 1, when there is one OFS, or the two OFS represent the same fuzzy term. Otherwise, as both OFS are neighbors, an extra area is accumulated at the top of the *y* axis to depict the reinforcement effect.

In our approach, reinforcement effect is used to know how the level of a concept's state is updated at time t_{i+1} by means of its own level and variation values held at t_i. It corresponds to a relationship level and variation to level (L+V-L).

3.3.7 Fuzzy-Causal Reasoning

Based on the earlier fuzzy concepts and procedures, Carvalo 2001 proposed additional elements to perform fuzzy-causal inference and outlined the mathematical foundations. Those items are also illustrated in Peña et al. 2008, 2012. Thus, we identify the key subjects involved in the fuzzy-causal reasoning as follows.

Firstly, fuzzy-causal relationships estimate how the concept's state is altered after a period. It means, a new variation value is computed at every increment of time. So the fuzzy term held by the consequent of a fuzzy-causal rule must be a variation regardless the antecedent's fuzzy term. Thus, two types of fuzzy-causal relationships are set: level to variation (L-V) and variation to variation (V-V).

Secondly, an output fuzzy causal set (OFCS) is computed to identify the fuzzy-causal effect that a cause concept produces on an effect concept. According the three sceneries, the OFCS own a MD of 1 and looks similar to the shape of the only one OFS, or the two OFS that corresponds to the same fuzzy term. Otherwise, an aggregation process is fulfilled to shape an "intermediate" fuzzy set with a MD of 1. The OFCS looks alike the involved neighboring OFS and meets the dimension criterion. Thus, the OFCS is more slanting to the OFS with the higher MD.

Thirdly, an *accumulative effect* is estimated to reveal the *fuzzy-causal effect* produced by several concepts on a given concept. Such an effect claims: The final effect should be higher than the highest effect depicted by the involved OFCS. The accumulative effect is represented as a displacement on the abscissa. Thus, an extra area is added to the fuzzy set by a carry function. As a result, a variation fuzzy causal set (VFCS) is made. Once the first OFCS is set as the current $VFCS_1$, it is accumulated with the second to outcome a new $VFCS_2$. The process is repeated with the remaining $VFCS_{3...n}$ until the final $VFCS_n$ is produced.

Fourthly, a defuzzification is carried out to specify a new uncertainty degree in the *x* axis. This value represents a kind of fuzzy-qualitative difference between increasing and decreasing variations computed by the effect concept. Afterwards a fuzzification is made to identify the new fuzzy terms with their respective MD.

3.4 Knowledge Acquisition

In order to provide adaptive education that meets learner's needs, educational systems need to know who she/he is. Usually, the response comes from the SM, which is a module that holds a student's profile to characterize user and an engine to manage learner's knowledge, acquire new one, and make inferences with it. Moreover, attributes to shape educational content need to be represented. Thus, content's profile is also tailored to describe the LO authored to teach the DK. The acquisition of knowledge to outline both profiles is explained in this Sect.

3.4.1 Development of the Student's Profile

In the experiment, where our approach was used, a sample of four domains was chosen to outcome a student's profile. The acquisition of knowledge about such domains was made by means of psychological and pedagogical tests. The idea is to obtain accurate and reliable information of the individual, that are widely supported by earlier studies made by experts of the field. Thus, the tools used to measure attributes of the four domains are the next (Peña-Ayala 2010):

- Learner preferences: The Gardner's Multiple Intelligence model (GMIM) is applied to measure eight kinds of learning preferences, such as: visual-spatial, verbal-linguistic, logical, intrapersonal, interpersonal (Gardner 1983).
- Personality traits: The Minnesota Multiphasic Personality Inventory (MMPI) designed by (Hataway and McKinley 1989) is used to analyze four scales:
  - Clinical: It owns ten attributes about mental health, like: social introversion, paranoia, hysteria, depression, psychasthenia, schizophrenia …
  - Content: It embraces fifteen attributes to shape and anticipate traits: fears, health concerns, anger, cynicism, antisocial practices, obsessiveness…
  - Supplementary: It encompasses eleven clinical attributes: anxiety, repression, ego strength, social responsibility, colleague maladjustment…;
  - Validation: It holds the next five attributes to measure the reliability of the responses: lie, inconsistency, variable, inconsistency, can not say.
- Cognitive skills: The Wechsler Adult Intelligent Scale (WAIS) measures two scales of skills to estimate the intelligence quotient (IQ) (Wechsler 2002):
  - Performance scale: It includes five skills: visual composition, visual-logical ability, visual skills, mental association, observation.
  - Verbal: It accounts six skills: auditory recall, comprehension, information, numerical reasoning, similarities, vocabulary.
- DK: We extended the Taxonomy of Learning Objectives (TLO) to identify seven levels of DK mastered by students according the next ascending series (Anderson and Krathwohl 2001):
  - Ignorance: Learner unknowns the DK subject.
  - Knowledge: Person recalls and knows the main idea about the topic.
  - Comprehension: Student understands the concept and identifies causes.
  - Application: Apprentice handles the concept and uses in other scenarios.
  - Analysis: Individual splits the object and recognizes hidden meanings.
  - Synthesis: She/he is able to generalize, predict, and outcome conclusions.
  - Evaluation: Trainee owns a criterion to judge the soundness of a subject.

3.4.2 Development of the Content's Profile

Due to the specific learning preferences, personality and cognitive skills held by a student, we authored several LO to teach a given DK concept. Thus, we account cognitive, pedagogical, and multimedia criteria to stimulate user to learn DK content. The aim is to identify which of the available LO is the most suitable to meet the personal characteristics of the student. Such a LO should encourage student as much as possible to fulfill the highest learning in comparison with others LO.

With the purpose to analyze how the educational content could bias the student's apprenticeship, we shape the content's profile of the DK. Such a profile embraces two levels, one to depict the concept to be taught, and another to describe the LO authored for being delivered to student. Both levels account the guidelines for using learning technologies with multimedia (GULTM) claimed by Guttormsen and Krueger 2000. The attributes to be considered are the next:

- At concept level: It reveals characteristics of DK concept such as: how abstract is the concept, complexity, degree of practical application, load of technical meaning embedded, how plentiful is the concept.
- At LO level: It depicts a LO from a dual perspective to mean a certain balance, like: pedagogical paradigm (Behaviorist, Socratic), used learning theory (e.g., Constructivist, Objectivist), degree of motion (static, dynamic), the level of verbalization (linguistic, non-linguistic), intensity of stimuli (sonorous, visual).

3.5 Knowledge Representation

Once the profile of the student and the content are outlined, their acquired attributes are stored as knowledge repositories. Later on, users (e.g., cognitive, pedagogue, knowledge engineers, and web designer) provide the semantic meaning of concepts, linguistic terms, UOD, and other items in an ontology. They also define the causal relationships and FCRB in the ontology. These activities are accomplished by the interaction between users and system's agents. Moreover, the organization and the administration of the repositories are carried out by the specialized agents, as the ontology agent, whose outcomes are shown in this Sect.

3.5.1 Repositories of the Student Model

Our CFSM shapes a mental model of the student through several domains. They offer key elements to draw inferences about the student's likings, behavior, weakness and strengths. The representation of the domains and other student's subjects is made by the use of repositories. The repositories contain facts and beliefs of a domain or subject. The statements are organized as an eXtended Markup Language (XML) documents. Such documents stores the next kind of content:

- Profile: It integrates the main attributes of the learning preferences, personality traits, and cognitive skills domains to shape a basic profile of the student. A sample of this sort of content is presented in Fig. 3.1.
- Complementary data: These repositories stores raw data, computed information, and detailed description of the attributes held in the domains (e.g., five repositories complement personality traits, four repositories enhance cognitive skills, and one repository extends learning preferences).
- Student's DK: It states the background and the current DK held by the student as a result of the lectures delivered by the educational system. An instance of the sentences used to depict the DK domain is introduced in Fig. 3.2.
- Assessment: It offers a record of the behavior, responses, and outcomes scanned during the sessions held by the student with the educational system.
- Personal: It owns student's personal data (e.g., name, age, academic degree…).

```
- <profile id_term="student_rofile" id_student="8">
-  <cognitive_skills id_term="cognitive">
-   <instance id_concept="comprehension">
     <real_level_value>2</ real_level_value>
     <fuzzy_term_level>too low</fuzzy_term_level>
     <fuzzy_term_level>low</fuzzy_term_level>
     <membersip_degree>0.4</ membersip_degree>
     <membersip_degree>0.6</ membersip_degree>
```

Fig. 3.1 A sample of the student's profile. It shows the evaluation made to the *comprehension*, a *cognitive skill*, held by a student, whose id is *8*. The measure reveals a real level of *2*, equivalent to *too low* and *low* fuzzy values with a MD of 0.4 and 0.6 respectively

```
- <profile id_term="student_profile" id_student="1">
-  <domain_knowledge id_term="dk">
-   <instance id_concept="law">
     <real_level_value>8</ real_level_value>
     <real_variation_value>6</ real_variation_value >
     <fuzzy_term_level>analysis</fuzzy_term_level>
     <fuzzy_term_level>increases little</fuzzy_term_level>
     <membersip_degree>1.0</ membersip_degree>
     <membersip_degree>1.0</ membersip_degree>
```

Fig. 3.2 An extract of the student's DK. It depicts the score fulfilled for the DK concept *law*, an item of the DK domain, that student, whose id is *1*, mastered. The level measure is *8*, equivalent to *analysis* with a MD of *1.0*; and the variation measure is *6*, similar to *increases little* with 1.0 as MD

3.5.2 *Repositories of the Content's Profile*

The content's profile contains two knowledge repositories: a *pattern* to depict the structure and attributes used for describing any LO, and an *instance* to shape a LO. Both repositories are stored as XML documents according to the next structure:

- Pattern: It sketches a structure to characterize content at concept and LO levels, as it has already stated in subSect. 3.4.2. In consequence, two sets of elements are considered to embrace elements devoted to depict a given attribute.
- Instance: It describes a LO authored according to specific cognitive, pedagogical, and multimedia viewpoints. Thus, it adopts the pattern structure and elements to provide the qualifications given to the attributes, as it is sampled in Fig. 3.3.

```
- <profile id_term="LO_profile" id_LO ="9">
-  <domain_knowledge id_term="dk">
-   <DK_concept_instance id_concept="law">
-    <Concept_criteria id_criteria="concept_level">
-     <attribute_instance id_instance="abstract">
       <real_level_value>7</real_level_value>
       <fuzzy_term_level>normal</fuzzy_term_level>
       <membersip_degree>1.0</membersip_degree>
      </attribute_instance>
:::
-    </Concept_criteria>
-    <LO_criteria id_criteria="LO_level">
-     <attribute_instance id_instance="dynamic">
       <real_level_value>10</real_level_value>
       <fuzzy_term_level>too high</fuzzy_term_level>
       <membersip_degree>1.0</ membersip_degree>
      </attribute_instance>
:::
     </LO_criteria>
    </DK_concept_instance>
```

Fig. 3.3 An example of the content's profile for an instance of LO, whose id is *9* to correspond to the DK concept *law*. It is split into two elements: Concept_criteria and LO_criteria, The first holds the attribute *abstract*, whose real level is *7*, equivalent to *normal* with a MD of *1.0*. The second contains the *dynamic* attribute and its real level is *10*, similar to *too high* and *1.0* as MD

3.5.3 Semantic Representation

Our approach organizes an ontology, to identify and define the meaning of concepts, causal relationships, FCRB, fuzzy terms, UOD, and other items. An ontology agent administrates the ontology (Peña-Ayala 2009). Essentially, the statement of the ontology is made through the use of the next four kinds of OWL sentences: *class*, *DatatypeProperty*, *FunctionalProperty*, and *class instances*.

Class defines a class and inheritance relationships with ancestors classes. Functional and data properties state properties and attach them to a class. Class instance creates an object of a class, whose properties have instantiated values. A sample of the first two OWL sentences is provided in Fig. 3.4.

```
00: <!–Statement of the class concept in lines 01-06 -->
01:  <owl:Class rdf:ID="relation" xmlns:rdf="rdf"…>
02:   <rdfs:comment rdf:datatype="…#string"…>
03:    Sets a causal link between a couple of concepts
04:   </rdfs:comment>
05:  <rdfs:subClassOf xmlns:rdfs="rdfs">
06:   <owl:Class rdf:about="#_id" />
:::
20: <!–Statement for attribute id_FCRB in lines 21-26 -->
21: <owl:FunctionalProperty rdf:ID="id_FCRB" xmlns:rdf="rdf">
22:  <rdfs:comment rdf:datatype="…#string"…>
23:   Identifies the FCRB that defines a causal relationship
24:  </rdfs:comment>
25:  <owl:unionOf rdf:parseType="collection">
26:   <owl:Class rdf:about="#_relation" />
```

Fig. 3.4 A piece of the ontology. Line 01 identifies the *relation* class, line 03 states the meaning, and line 06 links the class to the superclass *_id*. Line 21 reveals the *id_FCRB* attribute, whose definition is encoded in line 23, line 26 shows how to attach the attribute to the class *relation*

3.5.4 Representation of Causal Relationships

Causal relationships and FCRB are outlined in an ontology. Firstly, a class is defined for each item. Next the attributes to depict the classes are stated. Later on an instance class is edited for each relationship and FCRB, as it is shown in Fig. 3.5.

```
01:  <relation rdf:ID="visual-observation"…>
02:   <description…>visual bias observation</description>
03:   <id_FCRB xmlns:rdfs="rdfs">FCRB_51</id_FCRB>
04:   <id_cause_concept…">visual</ id_cause_concept>
05:   <id_effect_concept…">observation</ id_effect_concept>
:::
21:  <fcrb rdf:ID="fcrb_level5_variation11"…>
22:   <description…>FCRB level-variation</description>
23:   <id_cause_term…">low</ id_cause_term>
24:   <id_effect_term…">increase much</ id_effect_term>
```

Fig. 3.5 Ontological definition of a relationship (lines 01-05) and a FCRB (lines 21-24). In line 02 it is given the meaning of the relationship between the cause concept *visual* and the effect concept *observation*, whose id are respectively edited in lines 04 and 05. The meaning of the FCRB is presented in line 22, whereas lines 23 and 24 set a rule that claims: if the cause concept holds the fuzzy term *low*, then the effect concept is instantiated by the fuzzy term *increase much*

3.6 Cognitive Mapping

In a teaching-learning environment, a LO represents the *cause*, the student's mental faculties the *effect* (i.e., because of they are stimulated) and also the *cause* (i.e., due to they are used to learn), and the student's apprenticeship the final *effect* (e.g. a DK concept to be learned). In our work, this phenomenon is described by a CM.

Based on the workflow and algorithm set in Sect. 3.2, the Cognitive Mapping module automatically tailors the CM topology that corresponds to the LO to be evaluated. The cognitive mapping sketches the topology of the CM according to the concepts and relationships stated in the repositories of the SM, content model, and ontology. A description of the cognitive mapping process is explained as follows:

3.6.1 Architecture

The cognitive mapping is a process devoted, to sketch the topology of the CM that sketches how a given LO biases the student to acquire a DK concept. The topology embraces the next three tiers of concepts, where concepts bias each other by means of causal relationships (e.g., like a three-layer artificial neural network):

1. Cause tier: It holds cause concepts to reveal the attributes of the LO. It makes one arrangement to include concepts of both the already stated levels, concept and LO. What is more, a *6-level* UOD is attached to the concepts in order to instantiate their state. It owns the next fuzzy terms: {null, too low, low, normal, high, too high}. But, variation values are not used for these concepts, due to the content of LO which does not change during the provision of the lecture. Finally, these concepts only exert a causal bias on the concepts of the second tier.
2. Cause-effect tier: It encompasses cause-effect concepts to shape the student from several domains of view. It organizes an arrangement of concepts of the domains, that characterize the student (e.g., personality, cognitive, learning preferences). As these concepts are initially measured to shape the student's profile, the prior *5-level* UOD is attached to the concepts' state. In addition, these concepts are object of stimulation too. Thus, the already *11-variation* UOD is used to reveal, how the performance of a given concept is stimulated during the lecture. Concepts of the second tier exerts the concept of the third level and are influenced by it. They also bias each other and themselves.
3. Effect tier: It owns the DK concept to be taught (learned). The state of this concept is instantiated by the level and variation values, which were measured as prior DK. A *7-level* UOD with the next fuzzy terms {ignorance, knowledge, comprehension, application, analysis, synthesis, evaluation} is used to reveal the scale of the TLO mastered by the student for a given DK concept, as background DK and as current DK as well. Moreover, the *11-variation* UOD is used to show the tendency of the learning gained by the student. This concept is able to produce a causal influence on itself, and on the concepts of the second tier.

3.6.2 Process

The theoretical and mathematical baseline for tailoring a CM and make reasoning with CM as a tool for modeling system dynamics is deeply outlined in Peña 2008. But in this subSect., we resume the process as follows. As a first step, the mapping agent receives the *id* of the learner, the *id* of the concept to be taught, and the *id* of the LO option as parameters wrapped in a message.

Secondly, the causal relationships between concepts of the first and second tiers are set. They represent a kind of level-variation (L-V) relationship, because of the concepts' state of the first level only holds level values, and the consequences of the causal effect produces a variation on the second tier concepts' state.

Thirdly, concepts of the second tier bias the concept of the third tier, by variation-variation (V-V) and L-V relationships, due to they hold both values. Likewise, concept of the third tier exerts second tier concepts by similar relationships.

Fourthly, concepts of the second tier influence each other by V-V and L-V relationships. What is more, concepts of the second tier and the concept of the third tier update their level state value by means of L+V-L self-feedback relationships.

3.6.3 Outcome

A small sample of concepts and relationships involved to tailor a CM is chosen to explain how a CM topology is outlined, like the one sketched in Fig. 3.6, whose node's number corresponds to the id of the relation of concepts presented next:

At first tier, five concepts of the concept level are considered: 1) *abstract*; 2) *plentiful*; 3) *complex*; 4) *practical*; 5) *technical*) and eight concepts of the LO level are chosen: 6) *dynamic*; 7) *static*; 8) *constructive*; 9) *objectivist*; 10) *linguistic*; 11) *non-linguistic*; 12) *sonorous*; 13) *visual*). At second tier, four concepts are picked for each domain as follows: personality (e.g., 14) *hysteria*; 15) *psychasthenia*; 16) *social introversion*; 17) *depression*); cognitive (e.g., 18) *visual composition*; 19) IQ; 20) *auditory memory*; 21) *mental association*); learning preferences (e.g., 22) *visual-spatial*; 23) *logical*; 24) *verbal-linguistic*; 25) *intrapersonal*). At third level, it only has one concept, the DK concept to be learned, whose id is 26.

Thus, the topology of a CM is sketched by: 1) 156 L-V relationships (i.e., 13 * 12) to depict the bias between concepts of the first and the second tier; 2) 12 L-V and 12 V-V relationships (i.e., 12 * 1 + 12 * 1) to reveal how concepts of second tier bias the concept of third tier; 3) 12 L-V and 12 V-V relationships (i.e., 1* 12 + 1 * 12) as the feedback that concept of third tier outcomes on concepts of the second tier; 4) 132 L-V and 132 V-V relationships (i.e., 12 * 11 + 12 * 11) between concepts of the second level; 5) 12 L+V-L and 1 L+V-L self-feedback relationships for the 12 and 1 concepts of the second and third tiers respectively. In resume, the CM contains: 3 tiers, 26 concepts, and 481 causal relationships, whose nature is defined by their respective FCRB.

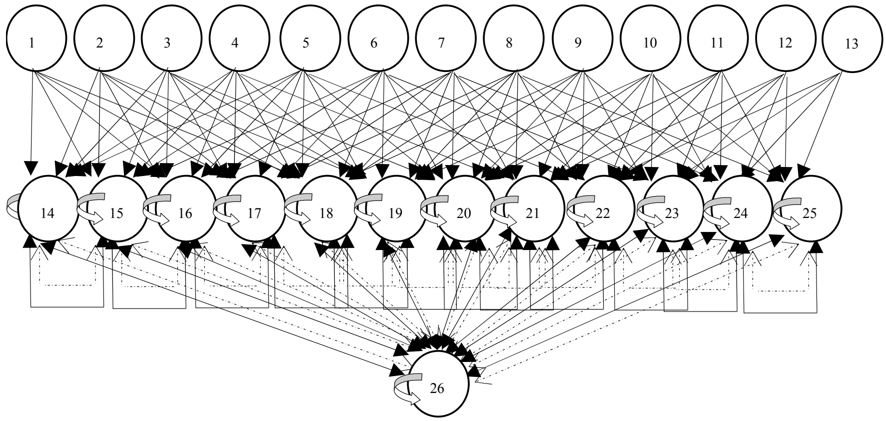


Fig. 3.6 A small version of the CM stated in subSect. 3.6.3 that is used in the experiment; where nodes depict concepts, the label is the *id* of the concept, arrowhead shows the target of the bias, continue line corresponds to *L-V*, doted line is a *V-V* and double-line states *L+V-L* relationships

3.7 Simulation

Our sequencing approach is acknowledged for its proactive functionality, as it makes decisions based on the anticipation of possible future events. Thereby, the student's apprenticeship is enhanced, because it is believed as the most promising LO, which is delivered to student at each teaching-learning cycle. Thus, the evaluation of the LO, authored for teaching a given concept is a matter of vital importance to accomplish such a functionality. The evaluation of the available LO is carried out by the simulation of the causal effects, that a LO exerts on the student's learning. Thus, a description of the simulation is given by: the identification of the inferences to be fulfilled and the algorithm used for making inference at concept layer.

3.7.1 Inference Flow

The simulation of the causal behavior and the outcomes that possibly happen during a lecture is essentially developed by means of the fuzzy-causal inference. This kind of reasoning estimates the fuzzy-causal impact that a concept produces on another. Besides the concepts and procedures earlier introduced in Sect. 3.3, we advice to consult the theoretical, mathematical, and graphical underlying items outlined for those subjects by Zadeh 1988, Carvalho 2001, and Peña et al. 2008, 2012. However, a workflow of the inference to be accomplished is drawn as follows.

Based on the criteria prior given, the next considerations are taken into account:

- The state of concepts in the first level never changes due to the content that is static.
- The level or/and the variation attached to the state of concepts in the second and third levels could change along the simulation when the concepts play the role of effect concept; otherwise, their state is static.
- Because of the learning gained by the student during the provision of the lecture, the concept of the third tier could feedback concepts of the second tier.
- The levels attached to the concepts' state in the first tier exert the state of concepts in the second tier as a variation estimated by L-V relationships.
- The variations occurred to the state of concepts, in second and third tiers bias the state of concepts in second and third tiers by V-V relationships.
- The levels assigned to the state of concepts in second and third tiers exert the state of concepts in second and third tiers by L-V relationships.
- The level and variation terms that instantiate the state of concepts in second and third layers bias their current level of their state by L+V-L relationships.

In addition, a pair of requirements must be met before the start of simulation. The first claims the sketching of the CM topology that represents the LO to be evaluated. The second demands the instantiation of the concepts' state according the measures estimated for the student and content profiles, set in Sect. 3.4. Considering the CM architecture tier, outlined in the subSect. 3.6.1, where the concepts are allocated, their state is instantiated by level or/and variation fuzzy terms.

Thereafter, the simulation of causal effects starts. Along discrete increments of time $t_1 \ldots t_n$, the causal effects on the *effect* concepts are estimated. Thus, a level or/and variation value is instantiated as the new concepts' state. The simulation ends, when a stable situation is reached (i.e. when the state of each concept does not change, a loop of values appears) or a chaotic situation is faced (i.e. it is evident the impossibility for reaching a stable situation after long time of process).

During each point of time t_i, the causal influence exerted on each concept of the CM is estimated, by the processes identified in subSect. 3.6 and 3.7 as follows:

- The variation of a concept's state, derived from L-V and V-V fuzzy-causal relationships, is the result of two steps:
  - The effect outcome by each fuzzy-causal relationship on a specific concept: It is computed by the generation of the next fuzzy sets: PFS, OFS, and OFCS.
  - The accumulative effect produced by several fuzzy-causal relationships on a given concept is estimated by the following processes: generation of the VFCS, defuzzification, and fuzzification.
- The level of a concept's state, stemmed from L+V-L FIR, is the outcome of the reinforcement effect exerted by each FIR on a specific concept. It is obtained through the next sequence of fuzzy sets: PFS, OFS, and IFS. This series is followed by defuzzification and fuzzification.

3.7.2 Inference at Concept Layer

Based on the criteria and procedures stated in Sect. 3.3, we shape the algorithm devoted to estimate, how the level or/and the variation of a concept's state is updated. Moreover in Fig. 3.7, it is pictured how the level of the state, that represents the concept to be taught evolves. It is the only one concept of the third tier.

```
00: Algorithm used to make inference at concept layer
01: Access ontological record of DK concept c to be taught
02: Access ontological record of fuzzy terms held by c
03: For each relation r_i where c is the effect concept do:
04:  Access ontological record of the relation r_i
05:  Access ontological record of cause concept ca of r_i
06:  Access ontological record of FCRB of the relation r_i
07:  Estimate the PFS_ca-c of the bias exerted by ca on c
08:  Estimate the OFS_ca-c based on the PFS_ca-c
09:  If (r is a fuzzy-causal relationship between ca and c)
10:    Estimate the OFCS_ca-c based on the OFS_ca-c
11: Estimate the variation of concept's state c, as follows:
12:  Set one array for increasing and one for decreasing OFS
13:  For each array, get a VCFS_c do:
14:   Set VCFS_c = OFS_1-c the current version to be accumulated
15:   For each OFCS_i-c from i = 2 to n do:
16:    Compare the variation revealed by VCFS_c and OFCS_i-c
17:    Displace the fuzzy set towards the one that holds the
       highest variation to outcome a new OFCS_c
18:    Compute an extra area by a carry function
19:    Add the extra area to the OFCS_c and create a new OFCS_c
20:    If the extended new OFCS_c exceeds x axis, then
21:     Estimate the overflow area
22:     Cut the extra area at both sides of the new OFCS_c
23: Defuzzification of the increasing and decreasing VCFS_c
24: Fuzzification to outcome the new variation fuzzy terms
     to update the state of concept c
25: Estimate the level of concept's state c, as follows:
26:  Set IFS_c = OFS_1-c the current version to be reinforced
27:  For each OFCS_i-c from i = 2 to n do:
28:   Estimate reinforcement effect between IFS_c and OFCS_i-c.
29:   Set the outcome IFS_i to be the new IFS_c
30: Defuzzification of the final IFS_c
31: Fuzzification to outcome the new level fuzzy terms
     to update the state of concept c
```

In lines 01 to 02, the ontology provides the semantics of the concept to be updated *c* and its current fuzzy terms. Next in lines 03 to 10, a cycle is achieved to process the fuzzy causal relations and FIR where *c* plays the effect role, as a result the PFS and OFS are the outcomes, besides the OFCS for fuzzy causal relations. Later on, lines 11 to 24 state how a new variation is computed for updating the state of *c* based on the procedure given in subSect 3.3.7. Lines 25 to 31 reveal the procedure set in subSect 3.3.6 to produce a new level fuzzy term that updates the state of *c*.

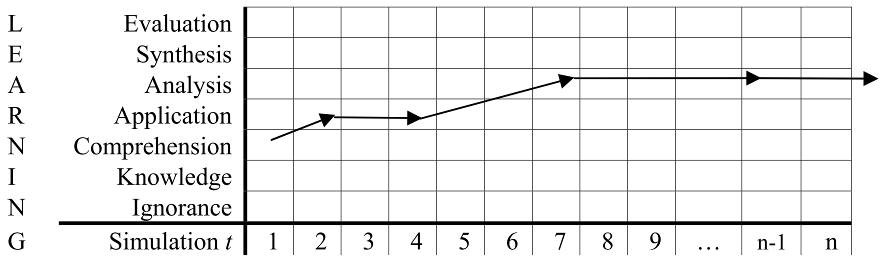


Fig. 3.7 A sample of a fuzzy-causal simulation of the bias that a LO exerts on the DK concept to taught. The abscissa depicts the discrete increments of time and the ordinate allocates the seven levels of the TLO. The consecutive arcs show how the concept's state evolves during the simulation until reach a fixed value. In consequence, the simulation comes to the end

It is drawn in Fig. 3.7 that at t_0, the student masters the *comprehension* level (i.e., prior DK of the concept). During discrete increments of time t_1… t_n the level evolves until reaching a stable level of *analysis*. Thus, this final level represents the expected bias that the evaluated LO could produce on the student's learning.

3.8 Experiment

In order to test our approach, we built three prototypes to implement the functionality of a WBES, a proactive sequencing, and also a FCSM. In addition, we designed a trial, with the aim at delivering proactive educational services to volunteers. The research hypothesis was: How the student's learning is improved when the WBES accounts a proactive sequencing based on a FCSM? A report of the experiment, statistical information, the interpretation of the results, and reflection of the findings are presented as follows.

3.8.1 Development

We accomplished a trial with two degrees for handling the independent variable (e.g., the student's apprenticeship): one with the provision of the stimuli (i.e., lectures of a DK, whose LO were chosen by our proactive sequencing) and other without the stimuli (the WBES randomly chose the LO delivered to students). In this way, we considered to recruit volunteers to organize two comparative teams: experimental and control. The stages of the experiment corresponded to: recruit participants, make psychological and pedagogical measures to shape a SM of each participant, training volunteers, pre-measure of prior DK, organization of the comparative groups, stimuli provision, and post-measure. Thereby, the main activities fulfilled during the trial are identified as follows:

- A web-based course about the "Scientific Method", as the DK, was authored to engage academic people to participate in the trial.
- A web-based campaign was launched to recruit volunteers of the country.
- Several hundreds of people were interested in participating in the trial and most of them filled the electronic form to apply as volunteers.
- The universe was made up by 200 volunteers, whose application was successfully filled. All of them were adults (i.e. their age ranged from 18 to 60 years old) and they held a Bachelor degree as the minimum academic level, or at least they were students at College.
- A series of four tests was applied to all volunteers according to the next sequence: learning preferences, personality traits, cognitive skills, and DK background.
- As a result of the measures achieved by volunteers, 50 people were picked to shape the population and continue participating in the trial.
- A training course about DK essentials was delivered to members of the population in order they became familiar with WBES and regulate the prior DK.
- Ten key concepts (TKC) of the DK were selected to shape the DK of the stimuli. They concerned to basic terms such as: hypothesis, law, theory…
- A sample of 18 volunteers was circumstantially organized. The criterion was to choose the first 18 participants that successfully accomplished the training.
- A pre-measure of the current knowledge held by volunteers about the TKC was applied to the sample. The degree mastered held by a participant was measured according to the seven levels proposed by the TLO.
- Two comparative groups of 9 members were randomly organized. The criterion was the consecutive order they got when the participant fulfilled her/his pretest. Thus, volunteers who ended in an odd position were assigned to the control team, the others to the experimental team.
- Four kinds of LO objects were authored from different cognitive, pedagogical and web-design perspectives to teach each of the TKC.
- Just one lecture about each TKC was delivered to volunteers. Thus, the WBES randomly picked up just one LO to teach the participant of control team. Whereas, volunteers of the experimental team were taught by a LO that was chosen by the WBES according to the advice provided by our proactive sequencing. Therefore, once the candidates LO were evaluated, the one that offered the highest student's apprenticeship was chosen.
- A post-measure of the new current knowledge held by participants about the TKC was made.
- Both pre and post-measures applied the seven TLO terms.
- The fuzzy terms used to reveal the level mastered by the volunteer for each TKC which were converted to a quantitative representation (e.g., 0 *ignorance*, 1 *knowledge*, 2 *comprehension*, 3 *application*, 4 *analysis*…).
- In consequence, a participant could achieve as *prior* or *new* DK for the TCK a total sum between [0, 60], and the prior or new DK sum of a team was [0, 540].
- Finally, the *learning gained* by a participant was considered as the difference between the new DK and the prior DK.

3.8.2 Statistical Analysis

Although the complete statistical outcomes are presented in Peña 2012, we provide a resume of the most relevant as follows: First of all, the assessment of the four psychological and pedagogical measures made to the members of the universe reveals: 113 participants successfully completed the learning preferences test, and only 102 of them also completed the personality exam. However, just 71 of the prior number fulfilled the cognitive measure, and only 50 of them rightly responded the background DK quiz too. Therefore, 75% of the universe deserted from the trial. As a result, the remaining 25% made up the population.

Concerning the sample n taken from the population, it was sized according the next criteria: a sample distribution standard error of 0.05, and a phenomenon occurrence probability of 0.92. As a result the size of n should be 18.62.

As regards with the characteristics of the comparative teams, in Table 3.1 a sample of key attributes is shown to reveal the balance achieved between members of both the control and experimental teams. It presents the distribution of the academic level and the number of formal occupations (e.g., job, study, sport…) held by the volunteer. It also offers three attributes of the learning preferences, personality traits, and cognitive skills domains to shape a common SM for both teams.

As a result of the pre and post measures, besides the learning gained by members of control and experimental teams, Table 3.2 outlines several statistical parameters, where it is evident control team held a former advantage about prior DK. However, after the stimuli provision, experimental team overcome the deficit and accomplished higher apprenticeship than control team.

The correlation between the prior and new DK was computed for both teams. Control team outcome a Pearson's coefficient r of 0.554, a statistical significance P of 0.122 to reveal a *positive* direction of the correlation and its intensity is *medium*, but, P is greater than the significance level ά of 0.05. Whereas experimental team achieved: $r = 0.828$; $P = 0.006$ to depict a *positive* direction, and *high* intensity of the correlation. Moreover, it is lesser than ά, thus it is so much significant!

In addition, a linear regression was estimated for testing the causal influence, that the prior DK and the stimuli exert on the new DK. The results for the pre and the post measures fulfilled by both teams are illustrated in Fig. 3.8 and interpreted next: Firstly, a ά of 0.05 was set. Secondly, according the linear regression formulae *Post* = *intercept* + *slop* * *pre-measure*, control team instantiates it as: *Post* = 13.7 + 1.22 * *pre-measure*; whereas, experimental team produces: *Post* = 7.72 + 3.28 * *pre-measure*. In consequence, it is interpreted control team owns a former advantage, due to its intercept is nearly two times greater than the one outcome by the experimental team. Regardless the deficit, it is confirmed experimental team overcame the disadvantage and reached a heavier bias on the post measure because of its slop is nearly three times greater than the one fulfilled by control team.

Moreover, in spite of experimental team held a lower prior DK than control team (e.g., 38 vs. 42 points), at the end they achieved a higher score (e.g., 198 vs. 174 points) and a better gained learning (e.g., 160 vs. 132).

Table 3.1 Descriptive statistics outcome for comparative teams

Criterion	Control		Experimental	
Maximum academic degree	Number	%	Number	%
- B. Sc.	2	22.2%	4	44.5%
- M. D.	6	66.6%	3	33.3%
- Ph. D.	1	11.1%	2	22.2%
Number of duties				
- 1	0	0%	1	11.1%
- 2	9	100%	8	88.9%
Learning preferences	Normal	High	Normal	High
- Verbal-linguistic	55.5%	33.3%	44.4%	33.3%
- Interpersonal	44.4%	44.4%	33.3%	33.3%
- Intrapersonal	33.3%	44.4%	33.3%	44.4%
Personality traits	Low	Normal	Low	Normal
- Health concerns	88.9%		66.6%	
- Maturity	55.5%		66.6%	
- Depression		66.7%		88.9%
Cognitive skills	Normal	High	Normal	High
- IQ	22.2%	0.0%	55.6%	44.4%
- Observation	66.7%	11.1%	22.2%	11.1%
- Verbal	33.3%	55.6%	22.2%	22.2%

Table 3.2 Basic statistics outcome from pre and post measures, and gained learning

Criterion	Control			Experimental		
	Pre	Post	Learning	Pre	Post	Learning
Sum	42	174	132	38	198	160
Mean	4.7	19.3	14.7	4.2	22.0	17.8
Range	15	25	22	9	29	26
Minimum	1	7	5	2	11	8
Maximum	16	32	27	11	40	34
Median	3	20	15	3	15	13
Mode	3	30	9	3	13	11
Standard deviation	4.4	9.7	8.2	2.8	11.5	9.3
Variance	19.8	95.0	66.7	7.9	133	86.9
Skewness	2.51	0.15	0.44	2.07	0.78	0.71
Kurtosis	6.9	-1.9	-1.2	4.7	-1.1	-1.0

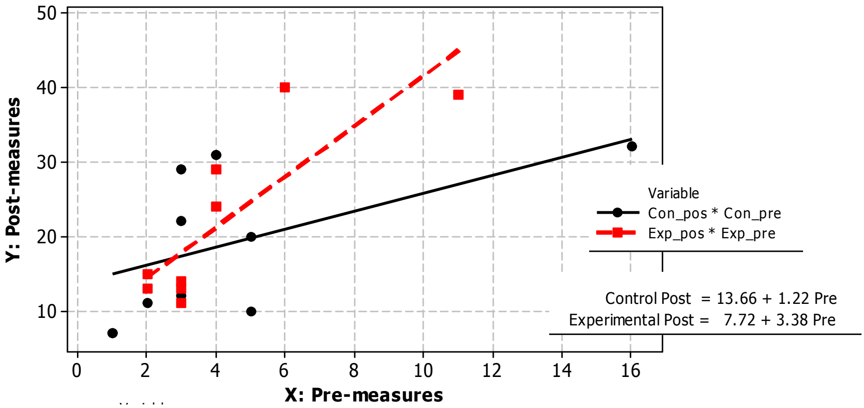


Fig. 3.8 A linear regression: It shows the dispersion diagram between Pre and Post measures for Control and Experimental groups

3.9 Conclusions and Future Work

Sequencing is responsible in choosing the best LO to meet student's needs according their interests, goals, strengths, and lacks. As regards with the SM, it aims at shaping a mental profile of the student's traits engaged at learning DK. Both modules are essential to enable the adaptive functionality of an educational system.

In this chapter, we focus on both sequencing and SM. Our work concerns the development of a proactive sequencing as a suitable paradigm to anticipate future outcomes and pursue to deliver the best available option to benefit student's learning. What is more, our approach included a FCSM as a relatively reliable source to know, who the student is and which strong mental attributes are candidate to be stimulated by the LO. In this way, we aim at enhancing student's apprenticeship.

As a result of the experiment, we collected empirical evidence to support the research hypothesis. Two essential arguments are resumed next. Based on the statistics already outlined, the team of students who were taught by LO recommended by our approach exceeded in 17.5% the learning gained by their peers that lacked of the proactive support. Furthermore, the cognitive skills domain revealed that none of the experimental members showed a high IQ level, but 44.4% volunteers of control team held such a level. Despite such cognitive disadvantage, experimental participants were able to accomplish a better score than those that were supposed to hold better intellectual resources.

As future work, we consider including more domains to shape the SM. Moreover, we plan to represent dynamic content that is self-adaptive during the provision of the lecture. In addition, we aim at implementing learning machine methods to interpret the behavior developed by student along the sessions. We are also interested in data mining algorithms to improve the accuracy of the predictions.

Acknowledgments. First author gives testimony of the strength given by his Father, Brother Jesus and Helper, as part of the research projects of World Outreach Light to the Nations Minis-tries (WOLNM). This work is supported by: CONACYT 118862, CONACYT-SNI-36453, CONACYT 118962-162727, IPN-SeAca/COTEPABE/144/11, IPN-SIP-20120266, IPN-SIP-EDI: SIP/DI/DOPI/EDI-0505/11, IPN-COFAA-SIBE.

References

Anderson, L.W., Krathwohl, D.R.: A taxonomy for learning, teaching, and assessing: A revision of Blomm's taxonomy. Longman, New York (2001)

Carvalho, J.P.: Rule base-based cognitive maps: Qualitative dynamic systems modeling and simulation. PhD Thesis, Lisboa Technical University (2001) (in Portuguese)

Duke, R., Madsen, C.: Proactive versus reactive teaching: Focusing observation on specific aspects of instruction. Bulletin of the Council for Research Music Education 108, 1–14 (1991)

Elsom-Cook, M.: Student modeling in intelligent tutoring systems. Artificial Intelligence Review 7, 227–240 (1993)

FIPA. Agent communication language. Technical specifications, Foundation for Intelligent Physical Agents (2000)

Hataway, S.R., McKinley, J.C.: Minnesota multiphasic personality inventory-2. Pearson, USA (1989)

IEEE-LTSC. IEEE 1484, Learning technology systems architecture. LTSA specifications, Institute of Electrical and Electronic Engineers, Learning Technology Standards Committee (1999)

Gardner, H.: Frames of mind. Basic Book Inc., New York (1983)

Guttormsen, S., Krueger, H.: Using new learning technologies with multimedia. IEEE Multimedia Journal 7(3), 40–51 (2000)

Merceron, A., Yacef, K.: Educational data mining: A case study. In: Proceedings of AIED, pp. 467–474. IOS Press, Amsterdam (2005)

Merrill, M.D.: First principles of instruction. Educational Technology Research and Development 50(3), 43–59 (2002)

Miguelena, F.: Scientific fundaments of the models. National Polytechnic Institute Press, Mexico (2000) (in Spanish)

Millan, E., Agosta, J.M., Perez de la Cruz, J.: Bayesian student modeling and the problem of parameter specification. British Journal of Educational Technology 32(2), 171–181 (2001)

NAVEDTRA: Task based curriculum development manual volume I. Developers guide, Naval Education and Training Command (1997)

Peña, A., Sossa, H., Gutierrez, A.: Causal knowledge and reasoning by cognitive maps: Pursuing a holistic approach. Expert Systems with Applications 38(1-2), 2–18 (2008)

Peña-Ayala, A.: Ontology agents and their applications in the web-based education systems: Towards an adaptive and intelligent service. In: Nguyen, N.T., Jain, L. (eds.) Intelligent Agents in the Evolution of Web and Applications, vol. 167, pp. 249–278. Springer, Berlin (2009)

Peña, A.: A student model based on cognitive maps. PhD Thesis, National Polytechnic Institute (2008) (in Spanish)

Peña-Ayala, A.: Acquisition, representation and management of user knowledge. Expert Systems with Applications 37(3), 2255–2264 (2010)

Peña-Ayala, A., Sossa-Azuela, H., Cervantes, F.: Predictive student model supported by fuzzy-causal knowledge and inference. Expert Systems with Applications 39(5), 4690–4709 (2012)

Sims, R., Dobbs, G., Hand, T.: Enhancing quality in online learning: Scaffolding planning and design through proactive evaluation. Distance Education 23(2) (2002)

Segravea, S., Holta, D.: Contemporary learning environments: Designing e-Learning for education in the professions. Distance Education 24(1), 7–24 (2003)

Self, J.A.: Deconstructionist Student Models in the Computer-Based Learning of Science. In: Diaz de Ilarraza Sanchez, A., Fernandez de Castro, I. (eds.) CALISCE 1996. LNCS, vol. 1108, pp. 27–36. Springer, Heidelberg (1996)

Self, J. A.: Formal approaches to student modeling. Technical report AAI/AI-ED 92, Lancaster University (1994)

Sevarac, Z.: Neuro fuzzy reasoner for student modeling. In: Proceedings of ICALT, pp. 740–744. IEEE, Los Alamitos (2006)

Timms, M.: Predicting students' need for help using pre-test data. In: Proceedings of AIED, pp. 1–6 (2001)

Wechsler, D.: WAIS test completo de inteligencia para adultos. Paidos, Argentina (2002) (in Spanish)

Wooldridge, M.: The logical modeling of computational multi-agent systems. PhD thesis, Manchester Metropolitan University (1992)

Zadeh, L.A.: Fuzzy logic. Computer 21(4), 83–93 (1988)

Zampunieris, D.: Implementation of a proactive learning management system. In: Reeves, T., Yamashita, S. (eds.) Proceedings of ELEARN, pp. 3145–3151. AACE, Chesapeake (2006)

Zampunieris, D.: Implementation of efficient proactive computing using lazy evaluation in a learning management system. International Journal of Web-Based Learning and Teaching Technologies 3(1), 103–109 (2008)

Zapata-Rios, M.: Sequencing of contents and learning, part II. Revista de Educación a Distancia 14, 1–15 (2006)

Abbreviations

ά	Significance level
CFSM	Causal and fuzzy student model
CM	Cognitive map
DK	Domain knowledge
DM	Data mining
FCRB	Fuzzy-causal rules bases
FIPA	Foundation for Intelligent Physical Agents
GMIM	Gardner's Multiple Intelligence model
GULTM	Guidelines for using learning technologies with multimedia
IEEE	Institute of Electrical and Electronic Engineers
FIR	Fuzzy inference relationships
IFS	Inferred fuzzy set
IQ	Intelligence quotient
ITS	Intelligent tutoring systems
LMS	Learning management system

LO	Learning object
LTSA	Learning Technology Systems Architecture
LTSC	Learning Technology Standards Committee
L-V	Level to variation
L+V-L	Level and variation to level
MAS	Multi-Agent System
MD	Membership degree
MMPI	Minnesota Multiphasic Personality Inventory
n	Sample
NAVEDTRA	Naval Education and Training Command
OFCS	Output fuzzy- causal set
OFS	Outcome fuzzy set
OWL	Web Ontology Language
P	Statistical significance
PFS	Preliminary fuzzy set
r	Pearson's coefficient
SM	Student model
TKC	Ten key concepts
TLO	Taxonomy of Learning Objectives
UOD	Universe of discourse
VFCS	Variation fuzzy causal set
V-V	Variation to variation
WAIS	Wechsler Adult Intelligent Scale
WBES	Web-based educational system
x	Abscissa axis
XML	eXtended Markup Language
y	Ordinate axis

Chapter 4
Exploiting Learner Models Using Data Mining for E-Learning: A Rule Based Approach

Marianne Holzhüter[1], Dirk Frosch-Wilke[1], and Ulrike Klein[2]

[1] University of Applied Sciences, Sokratesplatz 2, 24149 Kiel, Germany
{marianne.holzhueter,dirk.frosch-wilke}@fh-kiel.de
[2] University of Kiel, Zentrum für Geoinformatione, Boschstraße 1, 24118 Kiel, Germany
uklein@gis.uni-kiel.de

Abstract. The need for innovative didactical methods in combination with the efficient deployment of technical systems is an increasing challenge in the research field of e-learning. Research activities concerned with this observation have led to the understanding that the concept of learner models offers a range of possibilities to develop optimized, adaptive e-learning units (e.g. Graf et al. 2009, O'Connor 1998). Process information can enhance these approaches. Data mining is able to build process models from event logs. It means that information about real process execution can be deduced by extracting information from event logs rather than by assuming a behavior model which has been built by conventional modeling methods. This applies to the e-learning context, because a certain behavior of an underlying process model tracked in a Learning Management System (LMS) may differ from the one assumed by instructors or learning object designers of e-learning units. Instructors who need to attribute certain tasks to a huge group of online learners may not be capable of monitoring all factors influencing the appropriateness of all learner-task associations. Learning paths in LMS to which instructors have not paid attention to yet are of considerable interest. We apply a concept of rule based control of e-learning processes based on the framework we have presented in Holzhüter et al. 2010 to demonstrate these goals.

4.1 Introduction

Reflecting learner activities (especially to improve learning efficiency) in LMS gains importance as computer support of educational processes increases. The choice and deployment of LMS is associated with risks.

Especially if personalization is necessary, high implementation and maintenance efforts (Dagger 2006) and costs (e.g. Kleimann 2008) can be opposed to lack of user acceptance. Lack of acceptance is often caused by the absence of plausible arguments which convey the advantages for learners, instructors and LMS

A. Peña-Ayala (Ed.): Intelligent and Adaptive ELS, SIST 17, pp. 77–105.
springerlink.com

platform operators (see e.g. Tynjälä and Häkkinen 2005 for an illustration of the considerable range of challenges concerning e-learning projects). The improvement of e-learning process efficiency provides sound arguments to perceive the benefit potentials for learning and teaching scenarios.

The knowledge discovery discipline of data mining has developed to the extent that the extraction of useful information can be applied to several kinds of systems – including e-learning systems (Frias-Martinez 2006, Tynjälä and Häkkinen 2005). Data mining can efficiently support the generation of learner models. They support the adaptation of LMS to individual learner needs (Nguyen and Phung 2008). A particular type of LMS is explicitly dedicated this approach: Adaptive Educational Hypermedia Systems (AEHS).

The combination of these methodological and technological advances with findings from Business Process Management entails considerable potentials: It helps to focus on a learning process perspective exceeding the mainly isolated view on datasets which dominates in traditional data mining. Process mining, a specific sub-discipline of data mining, offers a useful set of concepts and tools to follow this learning process optimization and adaptation approach.

The chapter is structured by seven sections, partly further structured into subsections. The two next sections provide the terminological and theoretical foundations need to be laid. Sect. 4.4 is divided into two subsections: The first deals with our approach to combine the process mining and the learning style approach – as a method of learner modeling. The second subsection introduces the overall integration approach of rule based control in e-learning.

Sect. 4.5 is also split into two subsections: The firstly presents an architecture concept for our approach, ideally based on already existing technology in an institution (subSect. 4.5.1) and the second one deals with a possible implementation scenario based on open source software (subSect. 4.5.2*)*. The second approach is useful for institutions that are in the process of building up a completely new architecture and do not dispose of experience regarding the available technologies.

4.2 Related Work

As this research project partly is assigned to the geographical department of the University of Kiel, we especially focus on educational technology issues in the geographical field. Klein 2008 conducted a study which suggests that the discrepancy between employed media and the pupil's interest (in combination with assumed learning effectivity through the media of interest) reduces the quality of geographical teaching.

These two phenomena (discrepancy between media and interest and teaching quality issues) further were observed to reinforce each other. Especially schools (the target group of Klein's study) often lack means to employ media which pupils consider interesting and helpful for learning. E-learning help by supplying simulations, interactive maps or playful test trainings in a structured and efficient way of material presentation is desirable. This is where process mining in combination with learner models can provide a useful approach as we hypothesize. Learning processes can be identified and attributed to certain learning preconditions.

Favier and van der Schee 2008 proposed an exploratory study dealing with concrete learning paths. They discovered strong deficiencies in geographical thinking. Eventhough the approach provides hints concerning the way students are learning, we hypothesize that automatically building process models facilitates a better identification of learning problems. That is, because combining freedom of choice and the automatic generation of learning models reveals more influence factors than providing the learning paths (assumed as probable) and testing which students choose which alternative. E-learning process models can automatically be built by applying process mining to LMS (see subSect. 4.3.4).

Promising to enlarge the research and practitioners' community, the so called *ProM framework* currently qualifies as a process mining framework providing tools which represent the state of the art in the discipline (van Dongen et al. 2005). Although not the complete variety of tools is available in the framework, the researcher or practitioner disposes of tools applicable to educational processes.

Trčka et al. 2010 and Pechenizkiy et al. 2009 have worked on an approach to apply process mining to e-learning processes. They outline which process mining tools qualify best for supporting e-learning processes. As they deal with the complete range of tools which come into question they remain schematic about the process optimization scenarios involving further available technologies such as adaptation, learner modeling or rule-based control.

Romero et al. 2008 apply more methods from the knowledge discovery discipline of data mining to e-learning issues but do not refer to rule-based control which is of importance in a range of learning scenarios (see subSect. 4.3.1).

Cesarini et al. 2004 and Perez-Rodriguez et al. 2008 introduce different approaches for controlling learning paths with a workflow management system but do not apply a dedicated analysis strategy such as process mining. An appropriate analysis strategy enables instructors to identify the current e-learning processes with their optimization potentials.

Process mining belongs to the domain of BPM (Business Process Management) techniques which is a merely recent sub-discipline of data mining (van der Aalst et al. 2004). More precisely process mining belongs to the collection of tools in Business Activity Monitoring (BAM) and Business Process Intelligence (BPI) (van Dongen et al. 2005). Frosch-Wilke et al. 2008 have identified considerable connections between learning processes and business processes. Taking these connections into account approaches such as Helic 2006 or Cesarini et al. 2004 are of special interest, as they apply the concept of BPM to e-learning systems.

Grob et al. 2008 restrict their work to the business management domain. Consequently we exploited the connections between the business management approach and e-learning in Holzhüter et al. 2010.

Nguyen and Phung 2008 outline the benefit potentials derived from improving existing learner models especially in the context of AEHS. The process oriented perspective on learning suggests further that learner models need to be integrated into the learning process in AEHS (Adaptive Educational Hypermedia Systems, see subSect. 4.3.2). There are different aspects of learner modeling. Our approach combines the concept of learning styles with process mining methods. This combination results in a framework for exploiting learner models.

4.3 Basic Concepts

4.3.1 Control in E-Learning

In order to answer the question whether control of e-learning processes makes sense at all, consider the following argumentation. Restricting the freedom of choice in e-learning (as stated in Dron 2007) may seem contradictory to the well acknowledged goal of encouraging learners to be independent and autonomous individuals and help them develop a full repertoire of choice-making skills. Nevertheless, Moore 1980 opposes: A learner cannot learn effectively if the educational transaction demands more autonomy than he is able to exercise.

The argumentation for and against control in e-learning is not controversial but rather concerns different kinds of learning processes. The need for control in e-learning applies to the following category of learning processes: Processes with a rather uniform structure (Holzhüter et al. 2010).

If the learner is guided, he can make use of learning time more efficiently. If paths have qualified as effective for similar learners, this information helps to recommend a certain path for new learners with these similar preferences. This excludes learning activities where a high amount of creativity and variety in thinking is required and rather includes activities where it is necessary to precisely apply confirmed principles to a certain context, e.g. data modeling or logical reasoning. A learner may spend less time on tasks and reach the same outcome. In that way he obtains more time for tasks requiring his creativity and variety of thinking.

The findings of Lee and Wong 1989 militate for control in computer supported learning as follows: Learners with more freedom of choice were more active in dealing with tasks, but exhibited deficits in the post-tests regarding exact working.

In Hasselerharm and Leemkuil 1990 it has been demonstrated that adaptive control by a program had clear advantages over learner control (i.e. not the learner is controlled but rather he is the one who controls), but as well over non-adaptive program control.

Less control leads to more positive feedback of the LMS, but the final advice concerning this alternative is: Self-regulated learning needs to be taught first as a prerequisite. It is further not at all recommended related weak learners.

As illustrated in Kirschner et al. 2006, complexity reduction in LMS obviously needs to be considered. AEHS (see subSect. 4.3.4). It provides the necessary guidance to get a clear advantage over simple LMS, which are focused on freedom of choice. Kirschner et al. 2006 argue: the cognitive results can even be improved by AEHS if it is established and employed in the right way.

4.3.2 Learner Model

According to Zukerman and Albrecht 2001, learner modeling can be described as: Inferring unobservable information about a user from observable information about him/her (e.g., his/her actions or utterances).

As regards with Kobsa 1991, learner models are characterized in the following way: They represent assumptions about the types of learner characteristics (e.g. about knowledge, misconceptions, goals, plans, preferences tasks, and abilities). Learner models further represent common characteristics of learners regarded as relevant and which can be attributed to specific user subgroups of the application system (referred to as stereotypes). They classify users as pertaining to one or more of these subgroups and integrate the typical characteristics into the individual learner model. Learner models record learner behaviour.

Assumptions of the learner characteristics can thereby be derived from the interaction history of the learner with learning environments and can be drawn about current users based on initial ones. They can be justified by certain routines and the learner model can be maintained consistent in this way. The creation of stereotypes is possible by generalizing interaction histories (Kobsa 1991).

To explicitly relate this topic to another core topic of this chapter, note that the automatic generation of process models via process mining adds process related information to the static information usually supplied by the LMS. A way of supporting learner models in an e-learning context is to apply the concept of learning styles in combination with process mining.

Learning styles describe how learners perceive, interact with and respond to a learning environment; thereby measuring individual differences. We use the Learning Style Inventory (LSI) of Kolb (1984a, 1984b). Because of its elaborated theoretical foundation it belongs to the most intensely assessed approaches (Holman et al. 1997) concerning learning style concepts.

Kolb differentiates between four learning styles, which have been specified in Kolb et al. 1995. We have added findings regarding these learning styles from Lehmann 2010 which we apply in Sect. 4.4. Our choice is motivated by identifying preferences which clearly distinguish between students with a dominating the learning styles as follows:

- Accomodator (pragmatic focus): An accommodator prefers active experimentation and concrete experience. This learning style is especially supposed to prefer learning materials which relate to personal experience and which involve a high level of practical orientation, (i.e. involving a certain amount of interaction) (Jonassen and Grabowski 1993).
- Diverger (universalistic focus): A diverger prefers concrete experience and reflective observation. Jonassen and Grabowski 1993 have identified a need for reduction of practical orientation in learning scenarios for this learning style.
- Converger (specialistic focus): prefers abstract conceptualization and active experimentation. Staemmler 2006 and Bremer 2000 found that these learners prefer less interaction level/practical orientation and Jonassen and Grabowski 1993 recommend a deductive way of instruction. This means that instruction should start with the abstract and lead to concrete elements of learning content.

- Assimilator (theoretical focus): An assimilator prefers reflective observation and abstract conceptualization. Federico 2000 found that a practical orientation in e-learning platforms is of interest for this learning style. O'Connor 1998 found that personal experience unlike with accommodators reduces studying quality. Unlike convergers assimilators prefer inductive instruction or reasoning (Jonassen and Grabowski 1993).

The concept of learning styles has strongly criticized because of its lack of empirical evidence (Morgan 2008). Morgan 2008 states as well: "Learning styles clearly remain a field of further investigation as the range of useful findings may be integrated into current approaches". In what follows, we assume the learning style of a learner does not change throughout an e-learning unit (Lehmann 2010).

4.3.3 Business Processes and E-Learning Processes

The science of process management in the research field of business is more elaborated than in the research field of e-learning. Process analysis and management in e-learning qualifies as the more recent research field (Helic 2006, Cesarini et al. 2006, Trčka et al. 2010, Pechenizkiy et al. 2009).

Frosch-Wilke et al. 2008 have worked on identifying parallels between business processes and e-learning processes. These parallels help to classify e-learning processes.

Table 4.1 gives an overview of the concrete parallels and differences between the process types according to the business process characteristics in Stahlknecht and Hasenkamp 2005.

Table 4.1 Parallels between business process and e-learning process characteristics

Characteristic	Business Process	E-Learning-Process
Value adding	Value adding refers to the discrepancy between the output of an enterprise and the input to generate the output: a. business Process clearly is value adding	A learning process is not value adding in a monetarily measurable way but highly value adding for an individual learner.
Input and output	A business process always possesses an input and an output.	This characteristic is equivalent to that of a business process.
Timely order	Activities of a process can be executed in a parallel, a repeated and alternating.	This characteristic is equivalent to that of a business process.
Participants	A business process needs at least one participant. The participant does not need to be human.	This characteristic is equivalent to that of a business process.
Reusability	This characteristic refers to routine tasks which can potentially be standardized.	The reusability of an e-learning process is a prerequisite for successful process mining.

In addition, the categorization of business processes into production, support and management processes as in Becker and Algermissen 2007 is applied according to the next points, which are of special importance for the elaboration of our process mining framework for e-learning processes.

- The production or primary process represents the learning process. It is a key process among e-learning processes. It is closest to the production process.
- The support or secondary process corresponds to the support process for learning processes (mainly technical processes). A support process for e-learning processes is not directly value adding. Nevertheless, it is connected to the e-learning goals and might increase technical skills concerning an efficient interaction with the LMS interface. It is not related to specified learning topics.
- The management process corresponds to the moderation process. Leading and coordination of production and support processes executed by managers in a business company correspond to the action of leading and coordinating learning support processes executed by instructors.

4.3.4 Process Mining and E-Learning Processes

We use the concept of data mining to describe: "… the nontrivial extraction of implicit, previously unknown, and potentially useful information from data" (Fayyad et al. 1996, Frawley et al. 1991, Frias-Martinez 2006) and a process which identifies valid, potential, useful as well as understandable patterns in databases (Fayyad et al. 1996)".

The concept of process mining has firstly been formalized by van der Aalst et al. 2004. One of its goals is to build a process model in order to describe the behavior contained in event logs of information systems (de Medeiros et al. 2005). Event logs are the output of process-aware information systems (e.g., systems supporting workflow management). Fig. 4.1 summarizes the different process mining stages in a chart. The different components from Grob et al. 2008 have been applied to the e-learning context in Holzhüter et al. 2010.

It is necessary to distinguish three perspectives in process mining: 1) the process perspective is concerned with the ordering of events or the *control-flow*; 2) the organizational perspective provides information about the organizational field, which means how the different performers involved are related to each other; 3) the case perspective is not only concerned with the different paths in the process or the originators of a case, but as well with the values of the corresponding data elements (van Dongen et al. 2005).

To perform process mining, useful data categories need to be selected. First of all, the identification of process instances is necessary. An example concerning e-learning processes could be the process *learning unit about basic data modeling*.

In order to enhance the informative value of the process models, it is useful to enrich the mentioned instances by process objects. A process object could possibly be the learner. The process outcome refers to the process goals, in our case the understanding of the basic concepts of data modeling.

The increase of knowledge indicates to which extent this goal has been reached. It can be tested after the learner has finished the learning unit. The results are compared to a prior knowledge test. The process owner – in our case - corresponds to the learning unit. Process materials are ressources such as literature, audio explanations or simulations.

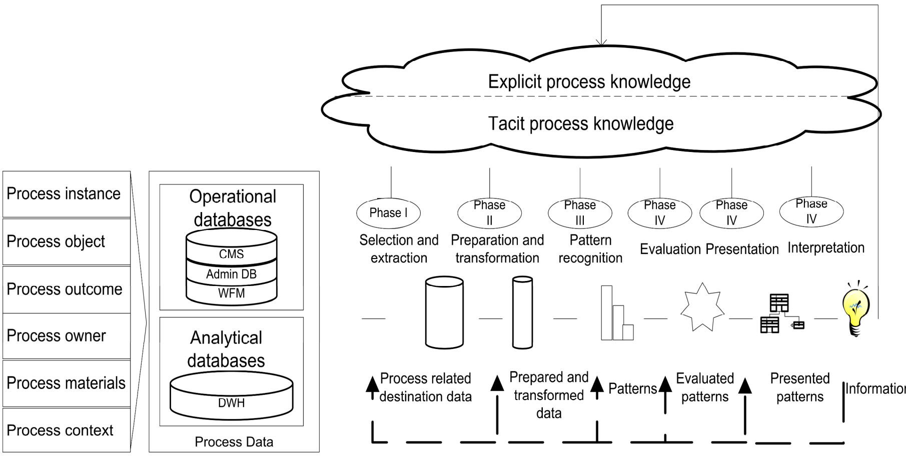


Fig. 4.1 Process mining framework (Grob et al. 2008, Holzhüter et al. 2010)

The spatial and timely elements are referred to as process context. These could be *start* and *end time* or *location*. Additionally, metadata may be considered for the documentation of the process execution.

Notice that Fig. 4.1 represents an ideal framework, which in its complete form rarely applies to common LMS. A Data Warehouse (DWH) or a Workflow Management System (WMS) may only be available anyway in business companies that use these kinds of technologies. This mostly does not only apply to qualification purposes but atleast to their business processes. In the genuine e-learning context, specific educational technologies are sufficient. The Content Management System (CMS) or the administration database (Admin DB or Administration in the next illustration) are rather common concerning e-learning environments. We refer to a selection of LMS technologies in the subSect. 4.5.2.

Further, data needs to be selected from the LMS databases. In order to carry out the analyses the selected data needs to be extracted, prepared and transformed. Pattern recognition follows. Before making use of the patterns an evaluation is made and after this step the results are presented and interpreted. This procedure improves explicit learning process knowledge, which is accompanied by implicit learning process knowledge that is gained unintentionally (Grob et al. 2008).

We introduced the concept of process mining in e-learning as well as that of learner models. In this aspect the term of AEHS needs to be mentioned as well – as process mining offers the potential to improve adaptive e-learning settings.

We refer to AEHS in the following way, considering the definition of Brusilovsky 1996: AEHS are hypertext, hypermedia or multimedia systems that include learner models and which are able to adapt the multimedia available in the system to these learner models.

4.4 Framework for Optimizing E-Learning Processes

4.4.1 Exploiting Learner Models Using Process Mining

Process mining in combination with the learning styles concept has been introduced as a promising method of learner modeling. The next sections pursue to demonstrate the framework behind this hypothesis as well as to present our empirical research, which currently consist in conducting a pilot study in a LMS.

A pilot study examining the surrounding conditions for process mining of e-learning processes is currently being conducted at the University of Kiel. Relating to the work of Klein 2008 the preconditions of effective learning in geographical subjects that may call for e-learning support are of special interest. Our current work is dealing with these kinds of preconditions, considering media competencies, experience with learning platforms and interest in (geographical) media.

The target group of this pilot study consists of geosciences students taking a course partly treating the topic of data modeling (which we refer to in our examples throughout the chapter) in semester 2011 at the University of Kiel. Data modeling is needed in the field of geographic information systems (GIS). A possible definition for GIS is based on Goodchild 1997: "A GIS is an information system to manipulate, summarize, query, edit, visualize (generally work) with information stored in computer databases. They use special information about *what is located where* on the earth's surface. Examples of applications are: Systems used by airlines and travel agents to manage reservations."

The discrepancy between employed media and the student's interest is as well analyzed and the results may supply arguments for the need of a certain amount of e-learning elements in the respective degree's program. The pilot study is in progress and not treated in detail in this book chapter.

Our approach is to combine the learning styles concept with process mining procedures. The empirical findings from the learning style research (see also sub-Sect. 4.3.2) provide general assumptions which are transferred to the geoinformation learning context. The assumptions in contrast to real life behavior in the learning environment are used to identify learning process elements which are critical to learning success.

The learning style preferences in Table 4.2 emphasize the differences between the learning styles. The numbers in brackets are necessary for illustrations further on in this section. Our goal is to improve e-learning units by building learner models that take the process-oriented perspective into account. Process mining is used to automatically generate process models. The advantage of building process models automatically is that frequent or low frequent learning paths are discovered according to real behavior tracked by an information system.

A test log of 30 process instances (one per originator) and 1,245 audit trail entries was generated to demonstrate which kind of learning behavior can be discovered by process mining methods.

Table 4.*2 Learning styles preference matrix*

Preference dimension	Diverger	Assimilator	Converger	Accomodator
1) Extent to which learning concerns personally	-	Preference: Avoid personal concern (0)	-	Preference: Increase personal concern (1)
2) Practical orientation	Preference: Reduce practical orientation (0)	Preference: Increase practical orientation (1)	Preference: Increase practical orientation (1)	Preference: Increase practical orientation (1)
3) Type of reasoning	-	Inductive (0)	Deductive (1)	-

Logs are rarely complete of free of noise. Consequently, it is important to dispose of algorithms that are robust regarding this incompleteness or noise. The Heuristic Miner algorithm – in contrast to the alpha++-algorithm - considers the frequency of traces in the log (Lang et al. 2008) and not only the relations between the activities *a* and *b* (van der Aalst et al. 2004). Other algorithms are as well resistant against noise but they are either more time consuming for processes with several activities or are not fully available in the framework used here: *ProM*.

For the learning process models of interest the Heuristic Miner algorithm meets the criteria needed. It mines the control-flow perspective of a process model and is robust against noise. Such an algorithm has been employed in other e-learning contexts such as in Pechenizkiy et al. 2009.

A learning unit has been employed in its testing phase by students studying geo sciences at the University of Kiel in summer semester 2011. It was used to generate the test logs. The learning unit was not possible to gather enough log files in the testing phase. Consequently we focus on an artificial test log to demonstrate the benefit potentials of process mining for e-learning.

Several learning paths were set according to the differences between learning styles (as they are represented in Table 4.2). One path is devoted to study the topic *data modeling*, where a student starts with the objectives or choose to think about the objectives on his own by a personal thought experiment.

Students choose between practical approaches or theoretical options. Exercises are offered to those who prefer interactive learning. Selected learning options are based on the preference dimensions of Table 4.2 as it is depicted as follows:

- The thought experiment against the simple presentation of the data modeling objectives (Dat_mod_objectives vs. Though_experiment in Fig. 4.3) depending on how much a learner prefers that the learning content concerns him personally (preference dimension 1 in Table 4.2). The task involves personal experiences of the learner. He is supposed to imagine how he can transfer the data modeling objectives to a scenario where he meets a person he likes, for instance: "*If you only know the first name what problem derives*?" (The person cannot be identified) or "*If you save the number in two cell-phones…*?" (You need to update the number twice in case it changes),

- Practical orientation in comparison more theoretical orientation (preference dimension 2 in Table 4.2). The alternatives are: The practical focus when dealing with the topic *normal forms* with examples against the theoretical focus with fewer examples and a better defined structure of explanation. Further, the preference of exercises is an indicator of a learner's practical orientation,
- If a certain type of reasoning is preferred, it is stated by the way a learner answers to the style the topic *normal forms* are shown. The learner comments the comprehensibility of the explanations. The practical explanations are built up following an inductive reasoning, putting examples at the beginning of the explanation. The theoretical explanations focus on the theory first and proceed to the examples later. The inductive reasoning in our e-learning unit implies examples are preferred as an introduction rather than theoretical statements.

The possibility to choose the learning style has no effect yet on the learning paths provided. This choice has been implemented to track learning behavior in relation to the respective learning style – which must be identified in advance. So that the process-aware information can be extracted from the LMS (in our case: Moodle) some mandatory information about an audit trail entry (the entries of the log file) in a learning management system is necessary to be able to create a file in Mining extended Markup Language (MXML) format, which is the data basis for process mining algorithms to generate process models as it is pointed out next:

```
<Process id="UNIFIED" description="Unified single process">
 <ProcessInstance id="0" description="Simulated process
    instance">
   <AuditTrailEntry>
    <WorkflowModelElement>
     login
    </WorkflowModelElement>
    <EventType>
     start
     </EventType>
    <Timestamp>
     2011-01T12:03:00.000
    </Timestamp>
    <Originator>
     S
    </Originator>
</AuditTrailEntry>
  </ProcessInstance>
</Process>
```

- The name of an event (e.g. a task in an online lesson),
- The start of an event,
- The end of an event,
- The timestamp, whereas the start of the next event may be the end of the preceding one, consequently the event type 'end' is optional,
- The originator, where the system or a specified user, in the example identified by a user identification number.

The dependency measure for simple sequencing is *a* and b stand for different learning activities or audit trails, which are described in detail by audit trail entries in a real log as it is stated in the MXML prior code and it is estimated by (4.1):

$$a \Rightarrow_w b = \left(\frac{|a >_w b| - |b >_w a|}{|a >_w b| + |b >_w a| + 1} \right). \tag{4.1}$$

It calculates how sure we can be about the dependency relation between *a* and *b*. $a \Rightarrow_w b$ is always between -1 and 1. $|a >_w b|$ indicates how often *b* directly follows *a*. A high $a \Rightarrow_w b$ value suggests that it is quite sure that *b* depends on *a*.

In order to identify loops, Weijters et al. 2006 additionally set the metrics. The simple sequence metric is always calculated first in the equations 4.2 and 4.3:

$$a \Rightarrow_w a = \left(\frac{|a >_w a|}{|a >_w a| + 1} \right) \text{ for length one loops} \tag{4.2}$$

$$a \Rightarrow_{2w} b = \left(\frac{|a >>_w b| + |b >>_w a|}{|a >>_w b| + |b >>_w a| + 1} \right) \text{ for length two loops} \tag{4.3}$$

In order to choose a certain reliability to be represented in the process model the dependency threshold can be adjusted. The calculation has been illustrated in equation 4.1. The default value in *ProM* is 0.9. An expert defines the thresholds for the process model, which is then generated by the heuristic miner algorithm.

It is of interest which dependency thresholds are necessary to discover whether certain behavior is due to noise or rather low frequent and might need to be considered when testing learning styles behavior.

In the test *log* at hand a dependency threshold of 0.6 is needed at least in order to eliminate the noise in the model (i.e. in our case caused by an interrupted internet connection of a few milliseconds) and further consider low frequent but relevant behavior as well. The result is shown in Fig. 4.3.

The process model can be utilized as follows: The conformance checker supplies the possibility to compare a priori models to a real life behavior in order to define an optimized a posteriori model (Rozinat and van der Aalst 2008). Transferred to the learning context with focus on learning styles, this implies: a priori models for the different learning styles can be compared to real life behavior of students (tracked in a LMS) with specified learning styles.

The priori models represent the findings from learning style research sketched in Table 4.2. Assuming that a process model regarded as reliable has been generated, following process instances could be compared with this model. We show a possible comparison of an assimilator learning path with the forecasted learning path (see Fig. 4.2). Highlighted events indicate a deviation from the a priori model. The short title of the task type, the answer value which applies (0 = alternative 1, 1 = alternative 2, no value= no task) and the event type (start or end) specify an event. The numbers in the input and output places (represented by circles) show how often a deviation has been registered.

Table 4.2 shows the preferences of the assimilator with the correspondent coding. Reasons for deviations are that personal concern in LMS differs completely from personal concern in traditional classrooms. Avoiding practical orientation could be caused by lack of technical skills in the e-learning environment.

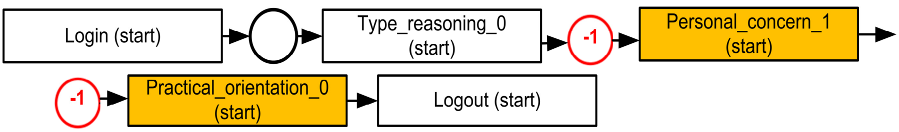


Fig. 4.2 Conformance check for the learning style of an assimilator

The filtering of less important events (for analyzing e-learning skills, one can abstract from tasks that do not require these skills) helps to detect the respective causal relations.

We now transfer these assumptions to the test log. It was generated in a way to demonstrate the following possible optimization scenarios: Although alternatives for more theoretical learning approaches as well as rather practical oriented content presentation have been provided, the conformance checker indicates when only one of the paths has been chosen. One can think of a group of students rather unfamiliar with the topic so that a more practical approach could seem more helpful. Still, the theoretical approach offers a logical structure, which is less 'interrupted' by examples and can be helpful for more students who prefer conceptualization in learning.

For example, if the last exercise (in our test learning unit: an exercise related to the topic *data types*) has never been done, possible interpretations could be:

- No student had the patience to work on this last exercise. All students preferred to end the learning unit,
- The importance of the topic *data types* could not be transmitted in a way that students would take the chance to improve their skills with this exercise.

But not only generally preferred learning paths can be discovered and analyzed. It is a further advantage of the process oriented perspective that it supplies information, which can explain studying success of different learners in the following way: Poor results are not solely attributed to lack of skills but can as well (e.g. in case a certain learning path has lead to these results) be caused by having suggested learning paths insufficiently (e.g. a learning path is recommended but not motivating enough so that, because learners are not obligated to follow the recommendation, it is not chosen).

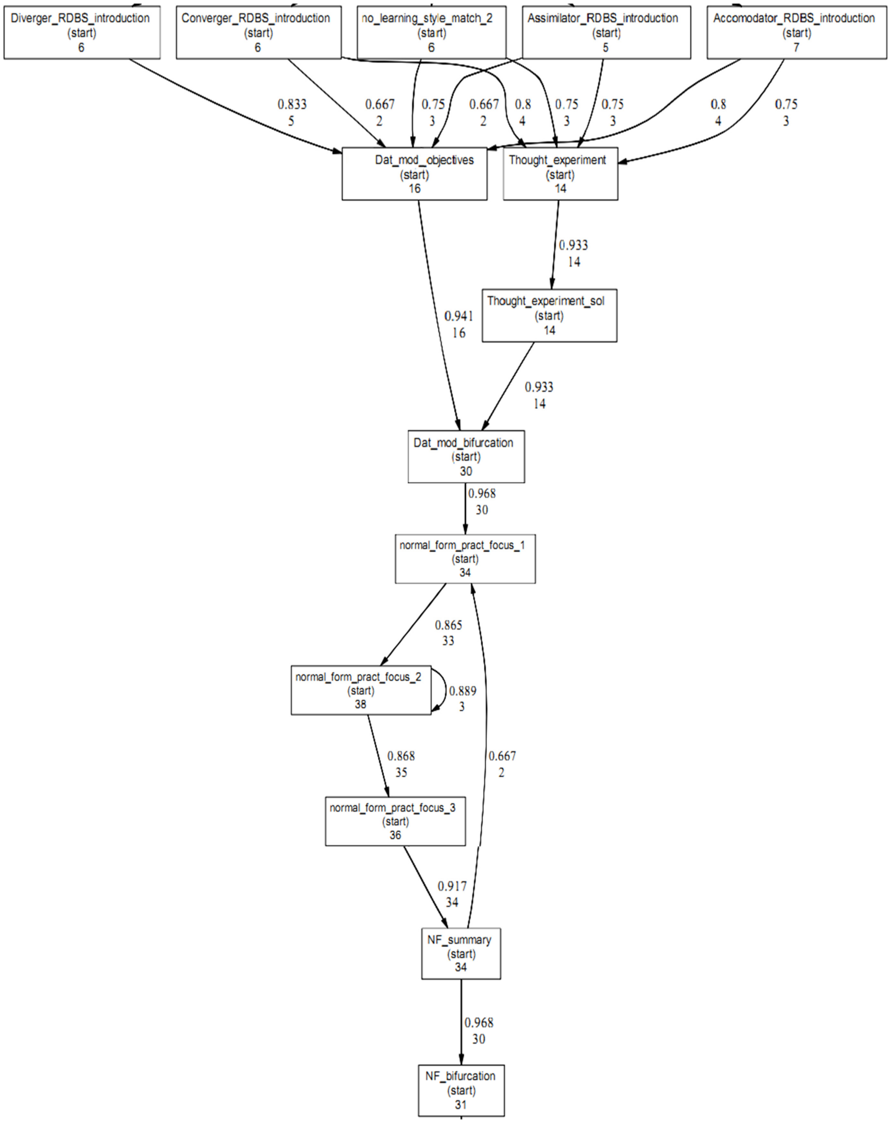


Fig. 4.3 Process model without noise

Provided the real life behavior has justifiably contradicted the a priori model (e.g. according to an instructors assessment), the learning environment is adjusted and the a posteriori model is supposed to represent the adjustments. Despite the automation it is necessary to subject the generated models to regular assessment by qualified experts.

The time students have worked on tasks can be related to their learning path and studying results. In *ProM* the different performance analysis tools can be employed for this purpose. Reasons for poor study results can be compared to the number and length of timely interruptions. When certain learning paths rather engender timely interruptions it might be interesting why this is the case (e.g. maximal attention span has not been considered for the e-learning unit or new concepts are included so that there is often need to catch up on the missing knowledge).

4.4.2 Integrated Approach

Having presented our process mining supported learner modeling approach, we now present the integrated rule-based approach which is rather suitable for institutions that already dispose of a DWH infrastructure and WMS. That often applies to enterprises with strong focus on process automation.

Assuming the crucial importance of learning in these kinds of enterprises (e.g. information technology related fields) the expected benefit of the rule based e-learning approach can be estimated as high (see Kirschner et al. 2006, subSect. 4.3.1). This is especially the case if the technology does not need to be implemented anymore but only needs to be adjusted to the respective requirements.

Provided the requirements to apply rule based control to e-learning have been met. That means we are dealing with learning processes which benefit from clear instruction opposed to freedom of choice (see subSect. 4.3.1) and the necessary technology is disposable. Then the procedure sketched in Fig. 4.4 shows how process mining and rule based control are used for monitoring learning processes and preventing ineffective process outcomes.

In order to apply rule based control a set of rules needs to be generated. The rule generation as the steps 1 - 3 of Fig. 4.4 requires already conducted mining activities from which a process expert can construct the relevant models (e.g., a decision tree). Our case expressed in a decision table is outcome from these models rules. In order to successfully improve processes, a normative action statement is generated on the basis of descriptive process knowledge.

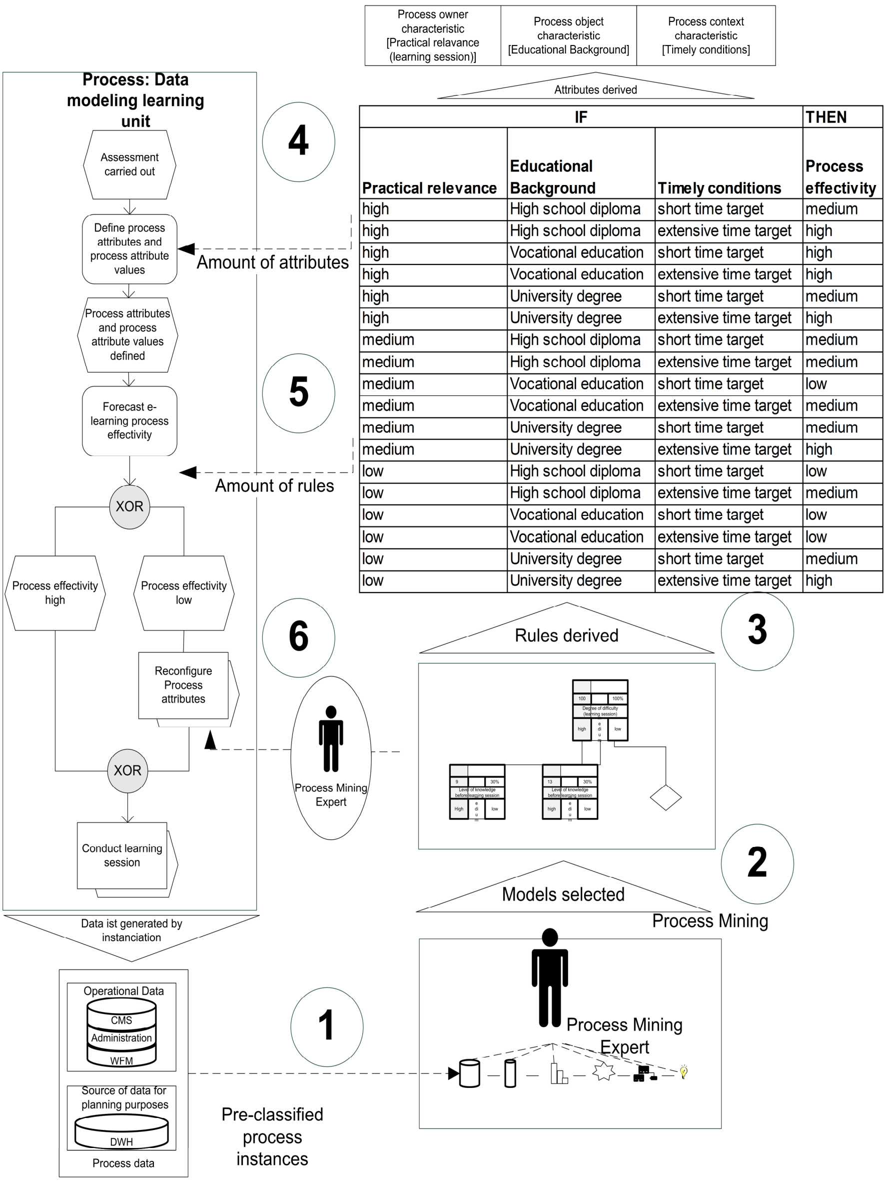
Note that we have chosen a rather generalist approach regarding the rules. In a real world scenario process characteristics are strongly context related. For illustration reasons, we chose a simple example for generating if-then rules.

The learner models, which can be automatically generated (e.g., as in Liu 2009 or built by simple questionnaires), are then supplemented by the process model information gathered from process mining.

To clarify the procedure we chose the scenario of a basic data modeling learning unit (as already applied previously) for learners without experience regarding this topic. The sample scenario is the following: A company plans to optimize e-learning processes in a trainee program with focus on data modeling skills. Trainees with different educational background are supposed to receive optimized data modeling training.

The next illustration demonstrates a useful combination of process objects. We chose: as a process object the learner with its process object characteristic *educational background* (high school diploma, vocational education or university degree); as process owner the learning unit with its process owner characteristic *practical relevance* (high, medium, low); as process context the time with its process context characteristic (i.e., *timely conditions* such as short time target).

Let the process goal be the following: The learner should practice data modeling by having increased his data modeling skills compared to the initial learning situation. So the skills are tested before and after the session.



IF			THEN
Practical relevance	**Educational Background**	**Timely conditions**	**Process effectivity**
high	High school diploma	short time target	medium
high	High school diploma	extensive time target	high
high	Vocational education	short time target	high
high	Vocational education	extensive time target	high
high	University degree	short time target	medium
high	University degree	extensive time target	high
medium	High school diploma	short time target	medium
medium	High school diploma	extensive time target	medium
medium	Vocational education	short time target	low
medium	Vocational education	extensive time target	medium
medium	University degree	short time target	medium
medium	University degree	extensive time target	high
low	High school diploma	short time target	low
low	High school diploma	extensive time target	medium
low	Vocational education	short time target	low
low	Vocational education	extensive time target	low
low	University degree	short time target	medium
low	University degree	extensive time target	high

Fig. 4.4 Integration approach for a rule-based control (Grob et al. 2008, Holzhüter et al. 2010)

As we focus on the left-hand side of Fig. 4.4, we assume the process characteristics are given and do not need to be extracted from decision trees or expert knowledge anymore. The learner is assessed, the process attribute values (values corresponding to the process characteristics) are stated and the process effectivity is forecasted. In case of ineffectivity, the process attribute values are reconfigured.

The process execution is documented and in that way can be used to improve future process effectivity forecasts.

We now categorize the different kinds of learning processes. The next model was presented by Holzhüter et al. 2010 in order to treat e-learning processes in analogy to business processes as in Grob et al. 2008 (see Fig. 4.5 and Fig. 4.6).

In order to systemize e-learning process types we apply a categorization of business processes within the dimensions *degree of structuring* and *process type* (Becker and Algermissen 2007). The categorization into process types has been explained in subSect. 4.3.3.

The dimension *degree of structuring* comprises categories depending on *the number and order of clearly defined and documented process steps* as well as *process outputs* to be attained. The classification contains three categories: *unstructured, semi-structured* and *highly structured.*

The more structured a process, the more suited it is for process mining procedures. This mostly requires clearly defined tasks, such as generating a data model, in contrast to tasks which can better be carried out with sufficient freedom of choice, such as the design of a poster.

The higher the possible contribution of processes for the e-learning goals (e.g., to gain a higher knowledge level or the reduction of learning time) and the higher the execution frequency, the more interesting they are for process mining guided optimization. The processes which do not sufficiently reach the required effectivity are selected from these.

In the selection model sample process instances 1 and 2 were selected for optimization, but the process efficiency of process instance 3 is similar to instance 1. As process instance 3 does not exhibit key deficits regarding process effectivity, it does not need to be optimised with the same priority as the other two.

Provided that an effective influence of variables on the process flow has been indentified via statistical methods (e.g., by use of regression analysis, distribution tests and scatter plots as well as the correlation between characteristics and the dependent variable using $\chi 2$ based methods) the variables have to be approved by an educational specialist regarding the respective processes. The identified correlations are important for the model generation, such as using decision trees.

Given the set of pre-classified objects it is possible to generate classification models to describe the classes and perform forecasts about new objects. The classification model can be used in a productive rule system within a LMS after the prognostic validity has been ensured (see the bottom of Fig. 4.6). Estimating the cross-validity (Breiman 1998) is a helpful method to comply this requirement, all of this is accompanied by the inspection of a process expert.

An important criterion for the prognostic validity is the data quality. The following criteria act as a toehold:

- Disposability, integrity, and consistency of the respective database,
- Integration in terms of logic, where data are unified in an relational schema,
- Appropriate timely reach (Grob et al. 2008).

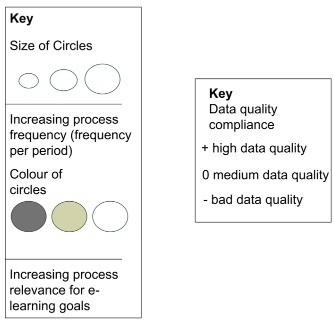


Fig. 4.5 Selection-model for choice of e-learning processes for rule-based process control

To assess and generate rules, it is crucial to analyze the correlations between the process characteristics and the extent to which the respective process goal has been reached. Table 4.3 shows an example of correlations between process characteristics, whereas we refer to the learning unit as process owner and to the learner as process object. The spatial and timely elements are referred to as process context. The process context comprises the spatial and timely elements of the process execution that constitute the starting conditions of the process instance.

The first line of Table 4.3 is depicted with the next example. The second line applies to a perspective where prior knowledge requirements depend on the learning location. This is explained as follows: Little prior knowledge of a learner opposed to high prior knowledge needs in public, where learning material is out of reach, leads to low process effectivity. Other cases are built in analogy.

Decision trees are normally based on real life process oriented datasets, which are to be extracted from different data sources and are then processed for

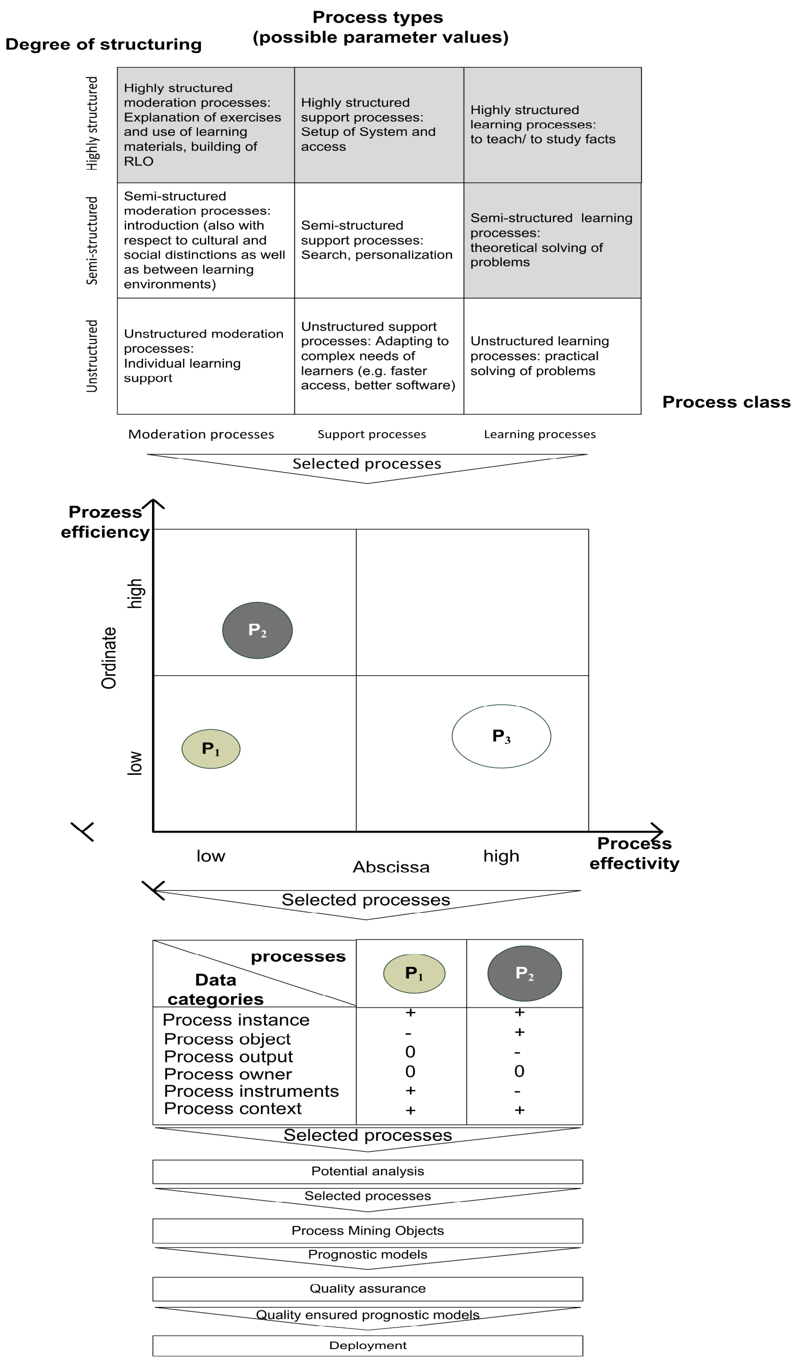


Fig. 4.6 Selection-model for rule-based process control (Grob et al. 2008, Holzhüter et al. 2010)

further use. The process mining expert accompanies this procedure and finally chooses the appropriate decision tree for the go-live of the system. An example of such a decision tree based on imaginary data is illustrated in Fig. 4.7 and Fig. 4.8. As in Grob et al. 2008, the decision tree is visualized in analogy to decision trees generated by the *Discoverer* tool of the prudsys AG. We calculated with a notional sample of 5,000 learners. We use such a huge sample to operate with reasonable amounts of instances when having reached the leafs on the top level of the decision tree. Regarding our illustration, it becomes clear how a decision tree can be built. As we assume correlations between the attributes, we could gather empirical evidence for rules which predict the effectivity of the learning processes.

It can be considered as a possible scenario, a company that plans to optimize e-learning processes in a trainee program with focus on data modeling skills as proposed in the introductory part.

Practical relevance is merely important for learners with vocational education, whereas university graduates can work quite well with a completely theoretical task. University graduates are not as good as learners with vocational education when the time target is very restricted and the application field is exclusively practical caused by lack of experience.

Table 4.3 Sample correlations between process attribute characteristics

Process owner characteristic	Process object characteristic	Process context characteristic
Practical relevance (high practical relevance, medium practical relevance, only theoretical relevance)	Educational background (high school diploma, university degree, vocational education)	Timely conditions (short time target, extensive time target)
Prior knowledge requirements (e-learning experience none, e-learning experience beg, e-learning, advanced e-learning experience, learning strategies, […]	Prior knowledge (no e-learning experience, little e-learning experience, medium e-learning experience, advanced e-learning experience, learning strategies, […]	PC access type (mobile, stationary) Access location (public, private)

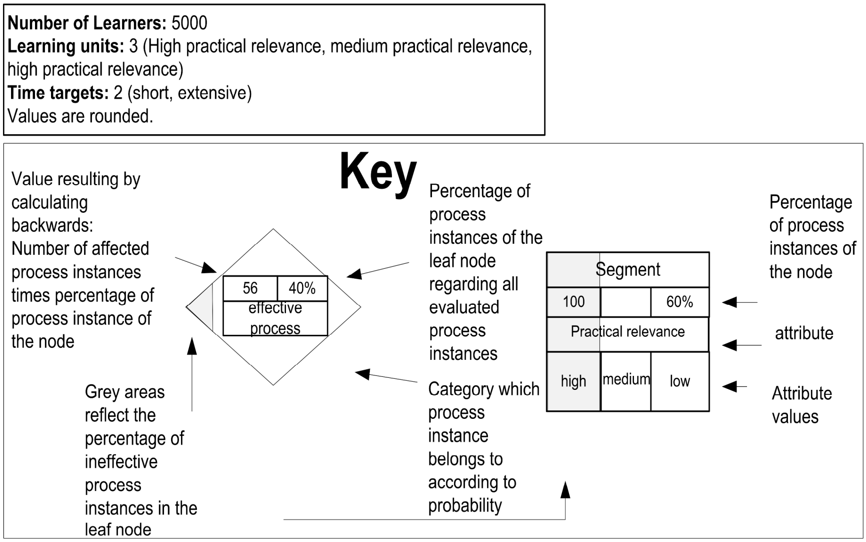


Fig. 4.7 Sample decision tree, legend

4.5 Implementation

4.5.1 Concept Integration into an Existing Architecture

We now introduce an implementation model of our process mining framework that is illustrated in Fig. 4.9. The following implementation architecture is intended as a general framework.

As a consequence we remain schematic about technical details. The following explanations go more into technical detail but do not consider the automation of rule-based control. The automation of rule based control remains a field of further investigation in the LMS development.

The process mining system hands over its models via Predictive Model Markup Language (PMML) interface to the rule management system. It is necessary that the rule management system is able to integrate, interpret the respective models, and access the necessary data. The process leading systems dispose of interfaces towards the process model and facilitate the determination of actual parameters of the attributes relevant to the forecast of the respective process instance to be executed. The operational data sources capture logs produced by the e-learning activities. The rules generated from the models are used to control the application. Dynamic interactions in the web are enabled by a web server. Instructors and learners are able to access the particular data in the database via Hyptertext Markuk Language (HTML) pages (Holzhüter et al. 2010, Cesarini et al. 2004).

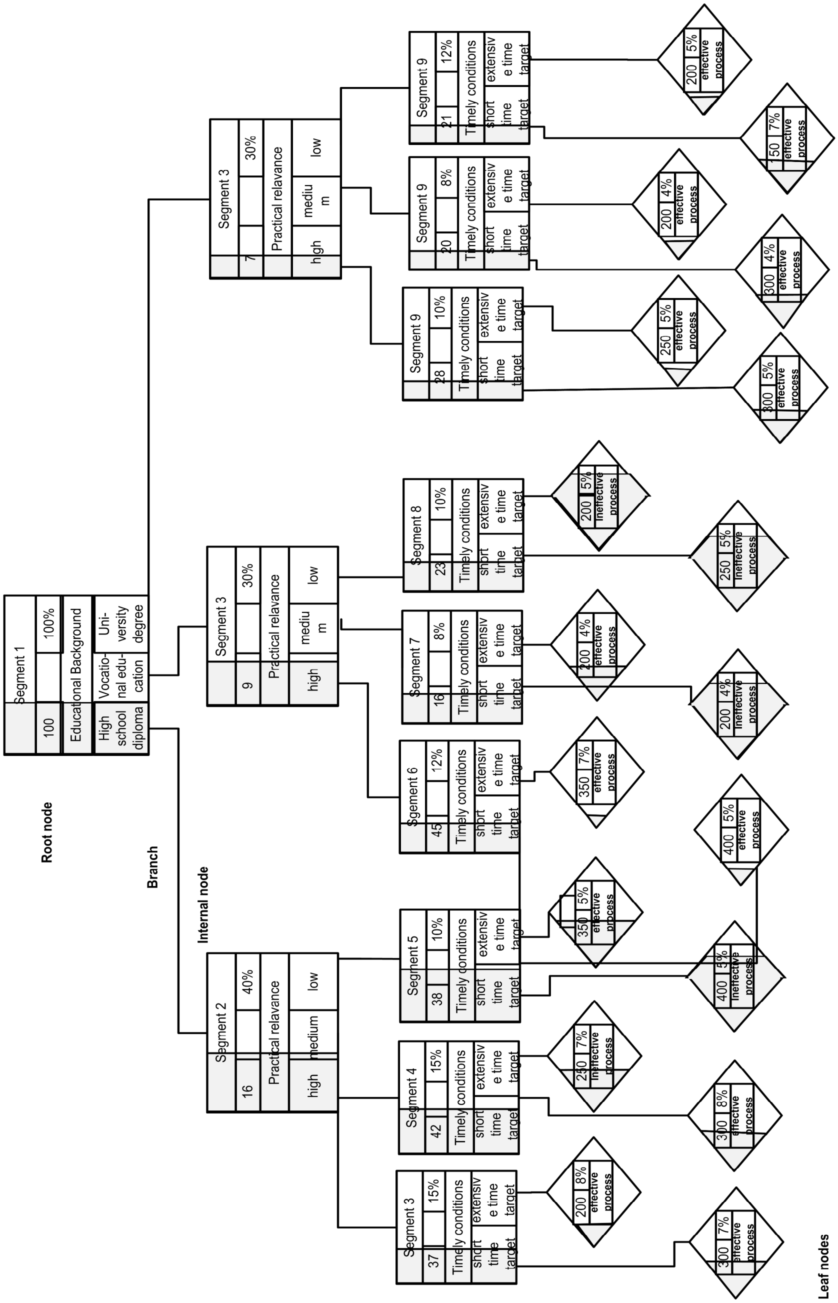


Fig. 4.8 Sample decision tree

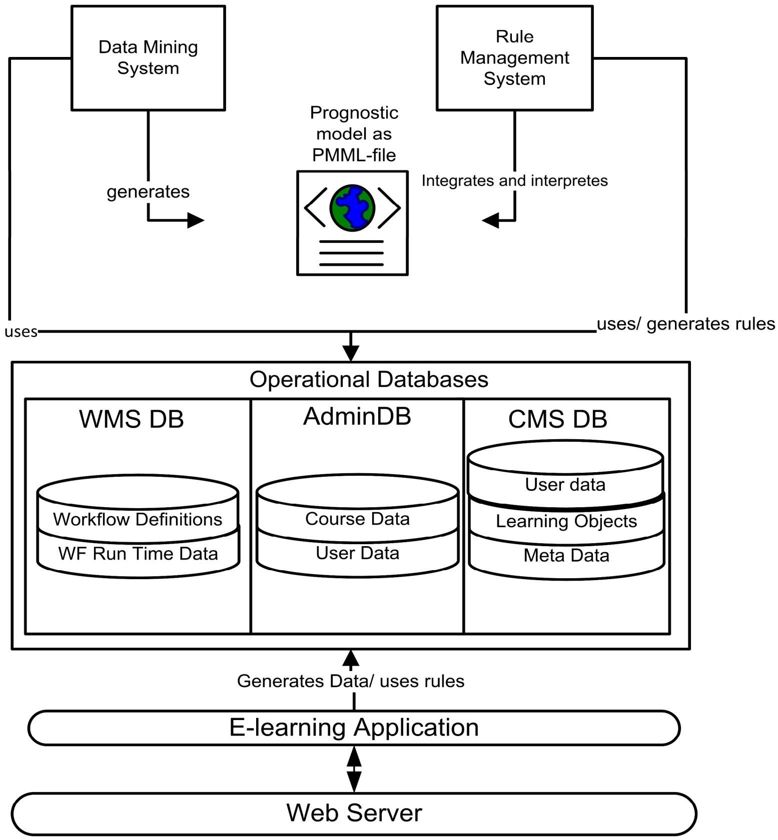


Fig. 4.9 Implementation (Holzhüter et al. 2010)

4.5.2 Implementation Scenario Based on Open Source Tools

The integration approach presented is recommended for e-learning scenarios in institutions where the infrastructure is already available. As will be discussed in our concluding remarks (Sect. 4.6), there is still a large amount of research to be conducted in order to implement workflows in e-learning environments, which uniquely serve the purpose of supporting e-learning processes.

We propose an approach for generating learning process models and present the available technologies. We focus on the open source LMS software Moodle. On server-side, Moodle is based on the Hypertext Pre-processor (PHP) a script programming language. On client-side, it works with Javascript and Cascading Style Sheets. Database technologies used are MySQL or PostgreSQL. For more detailed information read the online documentation (Moodledocs 2011).

First of all a priori learning process models for different learning styles need to be built. Expertise and/or decision trees are useful methods of gaining the necessary knowledge about learning outcomes, as it is shown in the step 1 of Fig. 4.10.

According to estimated appropriateness, the learning paths can then be implemented (Fig. 4.10, step 2). We propose an already developed approach which is presented in Rachbauer 2009. The appendix of Rachbauer 2009 contains the source code and necessary adjustments in the Moodle database, which help completely apply the concept of learning style based adaptation. The learners are then supposed to work with this LMS. Process mining of learning paths can now be carried out using ProM, whereas recommendations concerning further learning steps are given based on the a priori models.

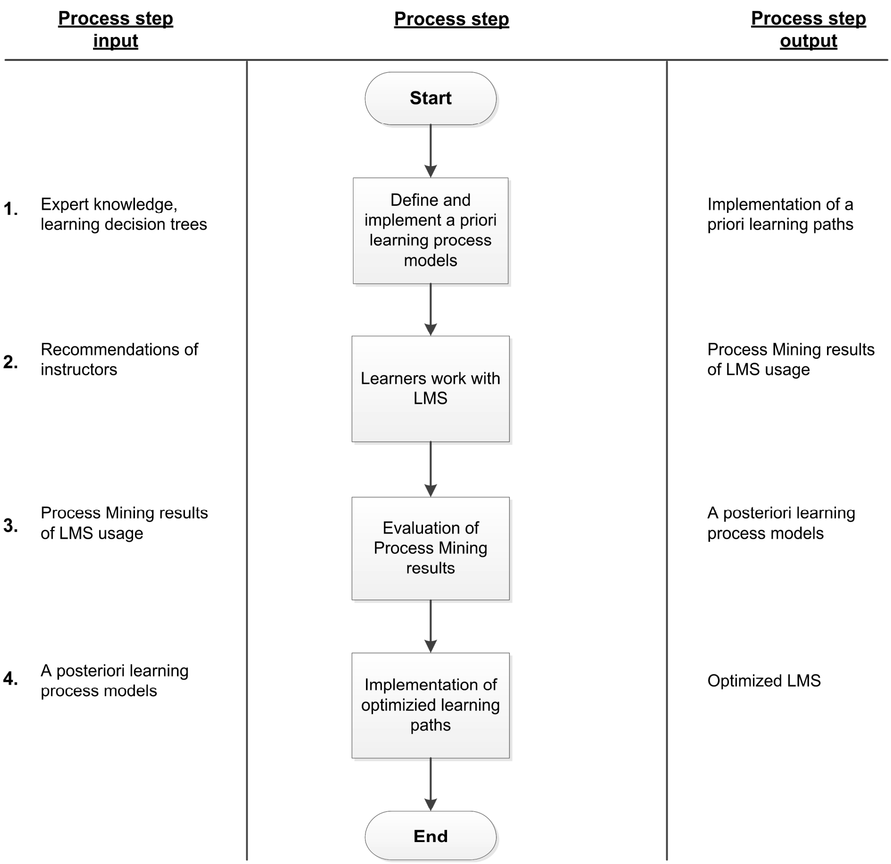


Fig. 4.10 Procedure of generating and utilizing process models

Finally the process mining results are evaluated, as it is sketched in the step 3 of Fig. 4.10, to improve a priori learning process models and turn them into a posteriori learning process models, in the way drawn in Fig. 4.10.

In order to extract the necessary information from LMS to build a priori learning process models, technologies are available in educational data mining tool suites such as KEEL (KEEL 2011) and WEKA (Bouckaert et al. 2010).

Romero et al. 2008 and Bravo et al. 2010 demonstrate how a decision trees can be generated and exploited. Learner models, which may partly have been build user guided or by automated methods as in Liu 2009 can as well be enhanced by this decision tree approach.

Expert knowledge and decision tree models given, different learning paths according to learner models can be implemented. Rachbauer 2009 gives a detailed explanation including the necessary source code for the realization in Moodle.

In order to apply process mining to Moodle, it is necessary to extract the necessary data from the LMS database and transform it into the MXML format. This can be carried out by a customized plug-in, written in Java, which can be developed with detailed guidelines from the online documentation (ProMimport 2011). Process mining can then be applied to mine learning paths in order to recommend further learning steps as well as to improve a priori learning process models.

4.6 Conclusion

Our argumentation was concerned with two research questions: 1) how can learning processes be optimized using process models and rule based control (such as via rule derivation from decision trees)? 2) how can process models in detail be generated taking into account the learning style concept?

Implementing rule based control (e.g. via workflows) into LMS rests an important field of further investigation. As Perez-Rodriguez et al. 2008 state: “the technology currently often employed, such as Moodle, must probably be completely remodeled to successfully apply workflow management”. Abdullah and Davis 2003 broaden this view to AEHS in general due to lack of standards and difficulty in using and reusing authoring material. They perceive a clear need of further development in order to successfully apply control in LMS.

The reason for splitting up the research topic only partly lies in a limited scope of this chapter. The main reason is that the complete integrated approach is rather addressed to e-learning in institutions already supplied with the respective technologies needed. The process analysis oriented approach without the rule based support leads to reasonable implementation effort by solely requiring already existing, freely available technology. This is of special interest for institutions that are clearly focused on education. For these institutions a negative cost benefit ratio is likely to result from completely remodeling an existing LMS.

An important aspect not to be neglected is that of privacy issues in LMS where learner behavior is tracked. In our research we feel strongly determined to respect learner privacy by not tracking learner characteristics and behavior in combination with complete names or other identifying information. Even if this issue is out of the scope of this chapter, it clearly needs to be mentioned and kept in mind.

Finally, e-learning process management and execution can be improved according to the findings of Lee and Wong 1989, Hasselerharm and Leemkuil 1990, Kirschner et al. 2006 – as we illustrated in subSect. 4.3.1. We provided a framework to design the respective implementation strategies. The improvement of e-learning processes is facilitated by the support of learning activities. These learning activities need to require structuring (i.e., it is necessary to precisely apply confirmed principles to a certain context such as data modeling or logical reasoning). A learner benefits of more time for tasks requiring his creativity and variety of thinking. That is, because he spends less time on the guided tasks.

Our approach can be used to find routines for improving adaptive learning environments which apply the learning styles concept. The aspects, which we have set, can be considered in empirical studies conducted to measure effects of LMS on acceptance and usage motivation of learners towards these environments.

This framework is part of a research project at the University of Kiel, geographical department and the University of Applied Science, department of Business Information Systems. A process mining prototype for e-learning processes competencies is being developed to deduce practical benefit potentials of process mining for e-learning processes based on empirical evidence.

Acknowledgments. The authors gratefully acknowledge comments received on earlier versions of this paper from Rainer Duttmann and Lasse Christiansen.

References

Abdullah, N.A., Davis, H.: Is simple sequencing simple adaptive hypermedia? In: Proceedings of the ACM HC, pp. 172–173 (2003)

Becker, J., Algermissen, L.: Prozessorientierte verwaltungsmodernisierung: Prozessmanagement im zeitalter von e-government und new public management. Springer, Berlin (2007)

Bouckaert, R.R., Frank, E., Hall, M., Kirkby, R., Reutemann, P., Seewald, A., Scuse, D.: WEKA manual for version 3-6-2. University of Waikato, Hamilton (2010), `http://www.cs.waikato.ac.nz/ml/weka/` (accessed July 23, 2011)

Bravo, J., Vialardi, C., Ortigosa, A.: Using decision trees for improving AEH courses. In: Romero, C., Ventura, S., Pechenizkiy, M., Baker, R.J.D. (eds.) Handbook of Educational Data Mining, pp. 365–376. Taylor & Francis, Boca Ration (2010)

Breiman, L.: Classification and regression trees. Chapman & Hall, Boca Raton (1998)

Bremer, C.: Virtuelles lernen in gruppen: Rollenspiele und online-diskussionen und die bedeutung von lerntypen. In: Scheuermann, F. (ed.) Campus 2000: Lernen in Neuen Organisationsformen. Waxmann, Münster (2000)

Brusilovsky, P.: Methods and techniques of adaptive hypermedia. User Modeling and User Adaptive Interfaces 6(2-3), 87–129 (1996)

Cesarini, M., Monga, M., Tedesco, R.: Carrying on the e-learning process with a workflow management engine. In: Proceedings of ACM Symposium on Applied Computing, pp. 940–945. ACM (2004)

Dagger, D.: Personalised e-learning development environments. PhD Dissertation, University of Dublin (2006)

de Medeiros, A.K.A., Weijters, A.J.M.M., van der Aalst, W.M.P.: Genetic Process Mining: A Basic Approach and Its Challenges. In: Bussler, C.J., Haller, A. (eds.) BPM 2005. LNCS, vol. 3812, pp. 203–215. Springer, Heidelberg (2006)

Dron, J.: Control and constraint in e-learning: Choosing when to choose. Information Science Reference-Imprint of IGI Publishing, University of Brighton (2007)

Favier, T., van der Schee, J.: Learning to think geographically with GIS. In: Donert, K., Wall, G. (eds.) Future Prospects in Geography, pp. 395–404. Liverpool Hope Press, UK (2008)

Fayyad, U.M., Piatetsky-Shapiro, G., Smyth, P.: From data mining to knowledge discovery in databases. AAAI MIT Press (1996), `http://www.aaai.org/ojs/index.php/aimagazine/article/view/1230/1131` (accessed December 24, 2011)

Federico, P.A.: Learning styles and student attitudes toward various aspects of network-based instruction. Computers in Human Behaviour 16(4), 359–379 (2000)

Frawley, W.J., Piatetsky-Shapiro, G., Matheus, C.J.: Knowledge discovery in databases: An overview. In: Piatetsky-Shapiro, G., Frawley, W.J. (eds.) Knowledge Discovery in Databases, pp. 1–27. AAAI Press/MIT Press, Menlo Park (1991)

Frias-Martinez, E., Chen, S.Y., Liu, X.: Survey of data mining approaches to user modeling for adaptive hypermedia. IEEE TSMC—Part C: Applications and Reviews 36(6), 734–749 (2006)

Frosch-Wilke, D., Sànchez-Alonso, S., Dodero, J.M.: Evolutionary design of collaborative learning processes through reflective Petri nets. In: Proceedings of IEEE ICALT, pp. 423–427 (2008)

Goodchild, M.F.: What is geographic information science? NCGIA Core Curriculum in Geographic Information Science. University of California Santa Barbara (1997), `http://www.ncgia.ucsb.edu/giscc/units/u002/u002.html` (accessed July 23, 2011)

Graf, S., Liu, T.C., Kinshuk, C.N.S., Yang, S.J.H.: Learning styles and cognitive traits: Their relationship and its benefits in web-based educational systems. Computers in Human Behaviour 25, 1280–1289 (2009)

Grob, H.L., Bensberg, F., Coners, A.: Regelbasierte steuerung von geschäftsprozessen-konzeption eines ansatzes auf basis des process mining. In: Die Wirtschaftsinformatik, pp. 268–281 (2008)

Hasselerharm, E., Leemkuil, H.: The relation between instructional control strategies and performance and attitudes in computer-based instruction. In: Pieters, J.M., Simons, P.R.J., de Leeuw, L. (eds.) Research on Computer-Based Instruction, pp. 67–80. Swets & Zeitlinger, Amsterdam (1990)

Helic, D.: Technology-supported management of collaborative learning processes. International Journal of Learning and Change 1(3), 285–298 (2006)

Holman, D., Pavlica, K., Thorpe, R.: Rethinking Kolb's theory of experiential learning in management education. In: Management Learning, pp. 135–148. Sage, London (1997)

Holzhüter, M., Frosch-Wilke, D., Sánchez-Alonso, S.: Discussion of the benefit potentials of process mining for e-learning processes. In: Proceedings of CSEDU, pp. 407–411 (2010)

Jonassen, D.H., Grabowski, B.L.: Handbook of individual differences learning, and instruction. Lawrence Erlbaum, Hillsdale (1993)

KEEL. Knowledge Extraction based on Evolutionary Learning (2011), http://www.keel.es/ (accessed July 23 , 2011)

Kirschner, P.A., Sweller, J., Clark, R.E.: Why minimal guidance during instruction does not work: An analysis of the failure of constructivist, discovery, problem-based, experiential, and inquiry-based teaching. Educational Psychologist 41, 75–86 (2006)

Kleimann, B.: Kapazitätseffekte von e-learning an deutschen hochschulen HIS. Hochschul-Informations-System GmbH (2008), http://www.his.de/pdf/pub_fh/fh-200806.pdf (accessed July 23, 2011)

Klein, U.: Geomedienkompetenz: Untersuchung zur akzeptanz und anwendung von geomedien im geographieunterricht unter besonderer berücksichtigung moderner informations- und kommunikationstechniken. PhD Dissertation, Geographisches Institut der Universität Kiel (2008)

Kobsa, A.: Editorial UMUAI: Special Issue on User Modeling Shell Systems. User Modeling and User Adaptive Interfaces 1 (1991)

Kolb, D.A.: Experiential learning: experience as the source of learning and development. Prentice Hall, Englewood Cliffs (1984a)

Kolb, D.A.: Learning styles inventory. McBer and Company, Boston (1984b)

Kolb, D.A., Rubin, I.M., Osland, J.: Organizational behavior: An experiential approach. Prentice Hall, Englewood Cliffs (1995)

Lang, M., Bürkle, T., Laumann, S., Prokosch, H.U.: Prokosch. H. U. Process mining for clinical workflows: challenges and current limitations. Stud. Health Technol. Inform. 136, 229–234 (2008)

Lee, S.S., Wong, S.C.H.: Adaptive program vs. learner control strategy on computer aided learning of gravimetric stoichiometry problems. Journal of Research on Computing in Education 21, 367–379 (1989)

Lehmann, R.: Lernstile als grundlage adaptiver lernsysteme in der softwareschulung. Waxmann, Münster (2010)

Liu, W.: Using data mining to dynamically build up just in time learner models. Master thesis, University of Saskatchewan, Canada (2009)

MoodleDocs (2011), http://docs.moodle.org/ (accessed July 23, 2011)

Moore, M.G.: Independent study. In: Boyd, R., Apps, J. (eds.) Redefining the Discipline of Adult Education, pp. 16–31. Jossey-Bass, San Fransisco (1980)

Morgan, C.: Experiential perspectives. In: Spector, J.M. (ed.) Handbook of Research on Educational Communications and Technology, pp. 33–37. Routledge, N.Y. (2008)

Nguyen, L., Phung, D.: Learner model in adaptive learning. World Academy of Science, Engineering and Technology 45, 395–400 (2008)

O'Connor, T.: Using learning styles to adapt technology for higher education. Indiana State University (1998)

Pechenizkiy, M., Trčka, N., Vasilyeva, E., van der Aalst, W.M.P., de Bra, P.: Process mining online assessment data. In: Proceedings of EDM, pp. 279–288 (2009)

Perez-Rodriguez, R., Caeiro-Rodriguez, M., Anido-Rifon, L.: Supporting PoEML educational processes in Moodle: A middleware approach. In: Proceedings of SPDECER (2008)

ProMimport (2011), http://www.win.tue.nl/processmining/promimport/start (accessed July 23, 2011)

Rachbauer, T.: Adaption von e-learning systemen: Moodle im vergleich zu anderen open-source-learning-content-management-systemen. Igel, Hamburg (2009)

Romero, C., Ventura, S., Garcia, E.: Data mining in course management systems: Moodle case study and tutorial. Computers & Education 51, 368–384 (2008)

Rozinat, A., van der Aalst, W.M.P.: Conformance checking of processes based on monitoring real behavior. Information Systems 33(2), 64–95 (2008)

Staemmler, D.: Lernstile und interaktive lernprogram: Kognitive komponenten des lernerfolges in virtuellen lernumgebungen. In: Gabler Edition Wissenschaft: Multimedia und Telekooperation. Deutscher Universitäts-Verlag, Wiesbaden (2006)

Stahlknecht, P., Hasenkamp, U.: Einführung in die wirtschaftsinformatik. Springer, Heidelberg (2005)

Trčka, N., Pechenizkiy, M., van der Aalst, W.: Process mining from educational data. In: Romero, C., Ventura, S., Pechenizkiy, M., Baker, R.J.D. (eds.) Handbook on Educational Data Mining, pp. 123–142. Chapman & Hall/CRC, Boca Raton (2010)

Tynjälä, P., Häkkinen, P.: E-learning at work: Theoretical underpinnings and pedagogical challenges. Journal of Workplace Learning 17(5-6), 318–336 (2005)

van der Aalst, W., Weijters, A., Maruster, L.: Workflow mining: Discovering process models from event logs. IEEE TKDE 16(9), 1128–1142 (2004)

van Dongen, B., Medeiros, A., Verbeek, H., Weijters, A.: The ProM framework: A new era in process mining tool support. In: Ciardo, G., Darondeau, P. (eds.) ATPN Nets 2005, pp. 444–454. Springer, Berlin (2005)

Weijters, A., van der Aalst, W.M.P., de Medeiros A.A.: Process mining with the heuristics miner-algorithm. Technical Report WP 166, BETA Working Paper Series, Eindhoven Univ. of Technology (2006)

Zukerman, I., Albrecht, D.W.: Predictive statistical models for user modeling. In: User Modeling and User Adaptive Interfaces, 11, pp. 5–18. Springer, Netherlands (2001)

Abbreviations

AdminDB	Administrative Data Base
AEHS	Adaptive Educational Hypermedia System
CMS	Content Management System
Dat. mod.	Data modeling
DWH	Data Warehouse
GIS	Geographical Information System
HTML	Hyptertext Markuk Language
LMS	Learning Management System
Moodle	Modular Object-Oriented Dynamic Learning Environment
MXML	Mining extended Markup Language
NF	normal form
pract.	practical
PHP	Hypertext Pre-processor or PHP Tools, Personal Home Page Tools
PMML	Predictive Model Markup Language
RDBS	Relational Database Management System
RLO	Reusable Learning Object
sol.	solution
WFM	Workflow Management
WMS	Workflow Management System

Part II
Content

Chapter 5
A Study of a Learning Style Index to Support an Intelligent and Adaptive Learning Systems

Mohamed Hamada[1], Kuseke Nishikawa[2], and John Brine[2]

[1] Graduate School, The University of Aizu, Aizuwakamatsu,
Fukushima, Japan
shadji@ee.duth.gr
[2] Direction of Primary Education of Eastern Thessaloniki
Katsimidi-Milou, 54638 Thessaloniki, Japan
{hamada,nishikawak,brine}@u-aizu.ac.jp

Abstract. An intelligent and adaptive learning system should adjust the content in order to ensure a faster and better performance in the learning process. One way is to help the learners and teachers to discover the preferences of learners. A learning style index is a method to classify the learning preferences of learners. Learning preferences can then help learners to find their most effective way to learn. It can also help teachers to adopt suitable learning materials for an efficient learning. This chapter is concerned with the study, implementation, and application of a web-based learning style index. We also describe a case study on the integration of the learning style index into an adaptive and intelligent e-learning system.

5.1 Introduction

An Intelligent and adaptive systems have established a long tradition in technology systems for individual learning. To better utilize such systems in the learning process, learners have to be aware of their learning preferences. A learning style index can help the learners to identify their learning preferences. It also supports to adopt suitable learning materials to enhance the learner's learning process.

On the other hand, teachers can gain by knowing their students' learning preferences. From the teacher's point of view, if they figure out their students' learning preferences, they can adjust their teaching style and adopt suitable materials to best fit with the students' preferences.

If there is a mismatch between a learner's learning style and the way learning materials are presented, students are more likely to lose their motivation to study.

Integration of learning style into learning systems can lead to an intelligent and adaptive learning system that can adjust the content in order to ensure faster and better performance in the learning process.

A. Peña-Ayala (Ed.): Intelligent and Adaptive ELS, SIST 17, pp. 109–132.
springerlink.com

So far, many learning models have been developed (e.g., Felder and Silver-man 1988, Herrmann 1990, Kolb 1984) for the realization of the learning preferences of learners. Among these models, the one made by Felder and Silverman 1988 is simple and easy to implement through a Web-based quiz system (Soloman and Felder 2009). Their model classifies learners into four axes: active versus reflective, sensing versus intuitive, visual versus verbal, and sequential versus global.

Active learners gain information through a learning-by-doing style, while reflective learners gain information by thinking about it. Sensing learners tend to learn facts through their senses, while the intuitive learners prefer discovering possibilities and relationships. Visual learners prefer images, diagrams, tables, movies, and demos, while verbal learners prefer written and spoken words. Sequential learners gain understanding from details and logical sequential steps, while global learners tend to learn a whole concept in large jumps.

While Felder-Silverman learning model was developed mainly for the engineering students, we think that with some modification, it can be adopted and used by junior learners at schools.

In fact, if the learners become aware of their learning style, it is not always true that their grades will improve. However, knowing their learning style can help learners continue to study. If learners can continue to learn something for a long time, gradually a gap widens between the learners who study based on their learning style and the others.

We found that the quiz system in Felder-Silverman "Learning Style Index" (LSI) allows students to chose between the two alternatives. However, in real life, not everything is black or white. Hence, freedom has to be given to learners to chose among several alternatives in a fuzzy-like system. Therefore we extended the Felder-Silverman LSI system to allow the students to choose among the five options.

Our *Enhanced LSI* (ELSI) is implemented in Java as an applet and integrated into a web-based system. The web-based system is connected to an SQL database using the Java Database Connector (JDBC). Using a database system is essential to analyze the ELSI for a group of learners and help the teachers to obtain a wider view of the learning preferences of their students.

To test and analyze our extended ELSI system, we applied the system with junior high school students and analyzed their learning preferences. Our system also distinguishes between male and female learners. This allows us to obtain a deeper understanding of the effect of gender differences on the learning process.

E-learning systems are widely used and rapidly increasing. The integration of LSI in an intelligent and adaptive e-learning system is useful to help e-learners to navigate through a different available learning materials. As a case study, we show the integration of our extended ELSI system into an intelligent and adaptive e-learning system for automata theory and theory of computation.

The rest of the chapter is organized as follows. Sect. 5.2 covers the background in learning systems, our study on the LSI, and the extension of existing systems. Sect. 5.3 covers our web-based implementation of the LSI. In Sect. 5.4 we apply our enhanced implemented system to the students from different high schools and junior high schools.

Then we analyze this result and report our observations. Sect. 5.5 describes the integration of our implemented LSI into an e-learning system for the theory of computation topics. We conclude this chapter in Sect. 5.6.

5.2 Learning Style Index

Among the existing learning systems we chose Felder-Silverman model for the following reasons:

- It is widely known and applicable,
- It can describe the learning styles in more detail than other models,
- Its reliability and validity have already been tested.

The Felder-Silverman LSI model classifies learners according to a scale of four dimensions: processing, perception, input, and understanding, as it is set in Table 5.1. Each of these dimensions consists of a contrastive attributes listed below.

The Index of Learning Style (ILS), based on the Felder-Silverman LSI model, is an outline questionnaire for identifying learning styles. The ILS consists of 44 questions for the afore-mentioned four dimensions, where each dimension has 11 questions. These preferences are expressed with the values between +11 to -11 and each problem has 1 or -1 (minus 1).

For example, if you answer a question related to ``active/reflective" attributes and your answer has an active preference, +1 is added to the score; whereas, 1 is subtracted from the score if you answer the question with a reflective preference. That is, the degree of preference for each dimension is just the algebraic sum of all values of the answers to the eleven questions, as it is shown in Equation (5.1).

$$i_{DIM} = \sum_{i=1}^{11} \mathrm{q_i}^{DIM} \tag{5.1}$$

Table 5.1 Learning and teaching styles

Learning Style		Teaching Style	
Process	Active Reflective	Student participation	Active Passive
Perception	Sensory Intuitive	Content	Concrete Abstract
Input	Visual Verbal	Presentation	Visual Verbal
Understanding	Sequential Global	Understanding	Sequential Global

Where, DIM is the set of dimensions that embraces four pairs of dimensions: {A/R, S/I, V/V, S/G} is the set of four dimensions, whose initial means: A/R for Active/Reflective; S/I for Sensory/Intuitive; V/V for Visual/Verbal; S/G for Sequential/Global. *I* is the vector of indexes composed by {iA/R, iS/I, iV/V, iS/G}. *I* describes the attributes in each dimension. *Q* is the sum of questions belonging to each dimension, thus *Q* ={q1, q2, … , q11}, and each q_i indicates the contribution given by the *i-th* question within the eleven questions for each DIM to detect whether preference 1 or -1 is substituted into q_i.

The results are divided into three groups, according to points shown in Fig. 5.1. If the score is between 3 and -3, the learner is categorized into "well balanced". If the score is between -5 and -7, or between 5 and 7, the learner is classified into "moderate preference". If the score is between -9 and -11 or between 9 and 11, the learner is grouped into "strong preference".

The reliability of LSI system was established in a western style educational institutes because the western style culture allows clear-cut "yes/no" answers for queries. On the contrary, the reliability of LSI is not clear in an Asian educational institutes because the Asian culture (especially Japanese) tend to permit unclear fuzzy answers for queries. Hence, in order to be able to study the reliability of LSI in an Asian educational institutes, it is necessary to extend the traditional "yes/no" style for answers to a new fuzzy-like system with an index of five levels. This extension will be explained in the next section.

5.2.1 Enhanced Learning Style Index

Our ELSI model extends the Felder-Silverman LSI model in two ways: a fuzzy-like evaluation system and a social/emotional dimension are introduced.

5.2.1.1 Fuzzy-like Evaluation System

Our model is based on answers of an ascending risk scale of 1 to 5 (Fig. 5.2). The assessment system extends the Felder-Silverman model as shown in equation 5.2.

$$i_{DIM} = \sum_{i=1}^{11} q_i^{DIM-} - \sum_{i=1}^{11} q_i^{DIM+} \tag{5.2}$$

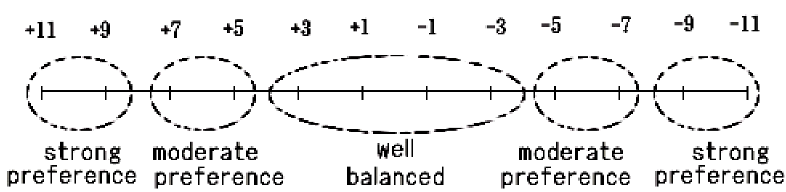


Fig. 5.1 ELSI evaluation system

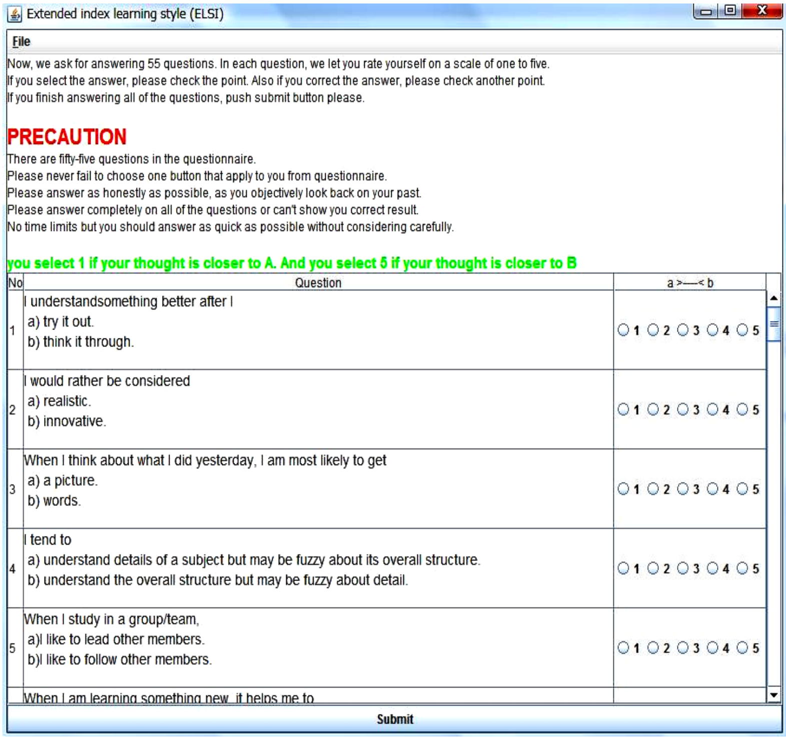


Fig. 5.2 ELSI five options system

DIM, *Q,* and *I* are the sets, previously set in section 5.2. q_i^{DIM+} and q_i^{DIM-} are attributes to represent the contrast of each dimension. The Felder-Silverman model only assigns *one* value 1 or -1 to each dimension when the learner answers a question. Our new model has *five* different values assigned for each question. Depending on the choice of the learner from the 5 scale values of the question's answer, q_i^{DIM+} and q_i^{DIM-} take one of the next positive or negative values: 1, 0.75, 0.5, 0.25, or 0, based on the next instances of selections made by the learner:

- S/he clicks 1st option in q_i, the value +1.0 is set to q_i^{DIM+} and 0 to q_i^{DIM-},
- S/he clicks 2nd option in q_i, the value 0.75 is set to q_i^{DIM+} and 0.25 to q_i^{DIM-},
- S/he clicks 3rd option in q_i, the value 0.5 is set to q_i^{DIM+} and 0.5 to q_i^{DIM-},
- S/he clicks 4th option in q_i, the value 0.25 is set to q_i^{DIM+} and 0.75 to q_i^{DIM-},
- S/he clicks 5th option in q_i, the value 0 is set to q_i^{DIM+} and 1.0 to q_i^{DIM-},

At the values assigned to the attribute q_i^{DIM+} and q_i^{DIM-} are accumulated. Then a subtraction between the two calculated values of the couple of attributes will be the result of learner's learning preference.

For example, suppose that the first choice is closest to “active” and fifth choice is closest to “reflective.” If learner choses the first option in this question, +1 point is added to the attribute of "active'. If learner selects the second option, +0.75 is added to the attribute of "active" and also +0.25 is added to the attribute of “reflective.” Likewise, if learner picks the third option, +0.5 is added to "active" and +0.5 to “reflective” and so on. Then the result of the learning preference in the active/reflective dimension is calculated by subtracting the total value assigned to “reflective" from that assigned to "active".

After the change in the point allocation system, we changed the degrees of preference (Fig. 5.3). If the learner’s score is between 11 and 7.5, or between -11 and -7.5, it is categorized into “strong preference.” If learner’s score is between 7.5 and 3.5, or between -7.5 and -3.5, it is classified into "moderate preference." If learner’s score is between 3.5 and 2, or between -3.5 and -2, it is grouped into “some preference.” If learner’s score is between -2 and 2, it is stated into "well balanced".

Several studies have been carried out to analyze the Felder-Silverman LSI model (Silvia et al. 2006, Silvia et al. 2007, Thomas et al. 2007), but none have considered the extension of the evaluation system in the way reported here.

5.2.1.2 Social/Emotional Dimension

Social emotional learning (SEL) is a process for helping people to develop the fundamental skills for achieving an effective life. SEL teaches the skills we all need to handle ourselves, our relationships, and our work, effectively and ethically. SEL holds the next five key:

- Self-awareness: assessing one’s feelings, interests, values, and strengths,
- Self-management: regulating one’s emotions to handle stress, control impulses, and persevere to overcome obstacles,
- Social awareness: is able to take the perspective of and empathize with others,
- Relationship skills: establishing and maintaining healthy and rewarding relationships based on cooperation,
- Responsible decision-making: making decisions based on consideration of ethical standards, safety concerns, appropriate social norms, respect for others, and likely consequences of various actions.

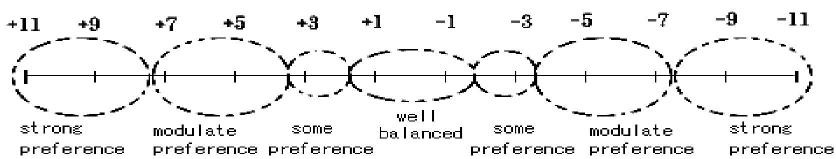


Fig. 5.3 Enhanced LSI evaluation system

These skills include recognizing and managing emotions, developing caring and concern for others, establishing positive relationships, making responsible decisions, and handling challenging situations constructively and ethically. SEL is a framework for school improvement.

Teaching SEL skills helps to create and maintain safe, caring learning environment. Social and emotional skills are implemented in a coordinated manner, school wide, from preschool through high school. Lessons are reinforced in the classroom, during out-of-school activities, and at home. Educators receive ongoing professional development in SEL. Families and schools work together to promote students' social, emotional, and academic success.

We extended the Felder-Silverman LSI model by adding a new "realistic" dimension that concerns with the effect of emotion and social learning styles. To this extent we added a new set of eleven questions to the quiz system of LSI for this new dimension. In designing these new eleven questions we referred to the Temperament and Character Inventory (TCI) model (Kumiko and Mari 2009).

Table 5.2 summarizes the new realistic (social/emotional) dimension, where the main attributes for both categories are the following:

- Social learners prefer reading books, discussions, social interaction, recognized and valued, and they may need repetition for detail,
- Emotional learners are affected by their emotion. Table 5.3 (based on Kort et al. 2001) represents a continuum of emotions ranging from positive to negative and their effect on learning. The emotions listed on the continuum can either affect learning in a positive (+) or negative (-) way.

5.3 Implementation

We built a web-based approach that embraces a web server, an application server and a database with the aim to analyze the learning styles. Some advantages of our model are outlined next.

1. Easy to use through its user-friendly interface,
2. Easy to integrate into E-learning systems. As we will explain in Sect. 5.5,
3. Easy to find and analyze the learning style of a group of learners. This enables the teachers to have a bird's view of the learning preferences of all students in the class,
4. Easy to access and use anytime anywhere.

The overview of our system is shown in Fig. 5.4. The system consists of the following components: a user-friendly graphical interface, a web-server, an application server, and a database module.

Table 5.2 Realistic (social/emotional) dimension

Realistic Learner	
Social	Emotional
Social learners tend to be big picture people; concepts are more interesting than details. They are motivated by relationships and care a great deal about what others think of them. They make more effort to attract people's attention. As a result, they are vulnerable to criticism. They also prefer cooperation rather than completion.	Emotions can affect the learning process, in both a positive and negative way. When a learner experiences positive emotions, the learning process can be enhanced. When a learner experiences negative emotions, the learning process can be disabled.

Table 5.3 Emotion sets possibly relevant to learning

Axis	-1.0	-0.5	0		0.5	1.0
Anxiety-Confidence	Anxiety	Worry	Discomfort	Comfort	Hopeful	Confident
Boredom-Fascination	Ennui	Boredom	Indifference	Interest	Curiosity	Intrigue
Frustration-Euphoria	Frustration	Puzzlement	Confusion	Insight	Enlightenment	Epiphany
Dispirited-Encouraged	Dispirited	Disappointed	Dissatisfied	Satisfied	Thrilled	Enthusiastic
Terror-Enchantment	Terror	Dread	Apprehension	Calm	Anticipatory	Excited

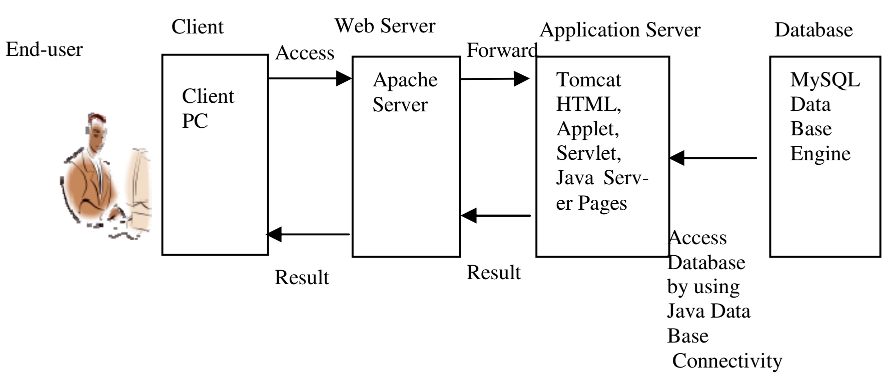


Fig. 5.4 Extended ILS system outline

The learning preferences computational module of our system resides in the application server. It uses the new calculation model described in subsect. 5.2.1. Such a model provides detailed information about the learning style of a learner.

The learner can access the system through the user interface. The system loads a java applet to run on a web browser (Figures 5.2 and 5.5). The learner then fills in all the answers of the quiz system and then submits the answers to the Apache (ASF 2008) web server through the client PC. The web server passes it to the Tomcat (Tomcat 2010) application server.

The application server runs the computational module of the system to estimate the learning preferences of the user. The application server sends the result back to the learner through the Apache web server and the Client PC. A copy of the result is also stored in the MySql database which is connected to the application server through the "Java DataBase Connector" (DBC). JDBC provides methods for querying and updating data in a database.

The system provides functions to maintain statistics with the learner gender distinguished. This helps educators to analyze the learning styles of their group students, even concerning the gender, and then prepare suitable teaching materials to adapt to their teaching style accordingly.

5.3.1 Web Application

Our web-based approach is designed according to the modular structure outlined in Fig. 5.6, where three main modules provide the basic functionality to different kinds of users as follows:

- The students' module enables an individual student to analyze his/her learning preferences and/or send them to his/her teachers,
- The teachers' module enables teachers to access and analyze their students learning preferences individually or in groups, male or female sets, and get a graphical representation of their students learning preferences,
- The administrator module maintains the system and the data base.

5.3.1.1 Students' Module

An individual student can access the system through the interface shown in Fig. 5.7. Then the student can answer the questioner and have her/his learning preferences analyzed automatically by the system. If the student provides her/his "Student ID", the system will store her/his learning preferences in her/his teachers' data base.

Fig. 5.5 Extended ILS web-based interface

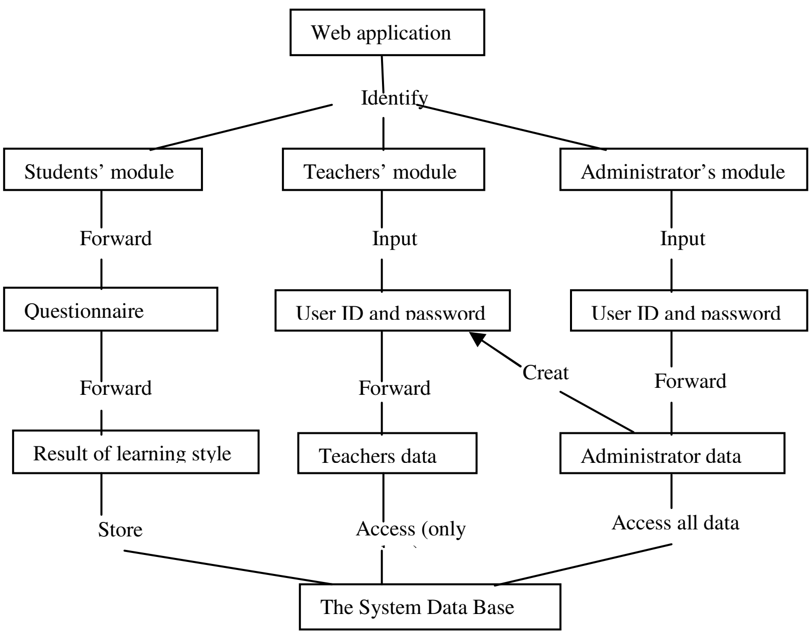


Fig. 5.6 The Web-based application architecture

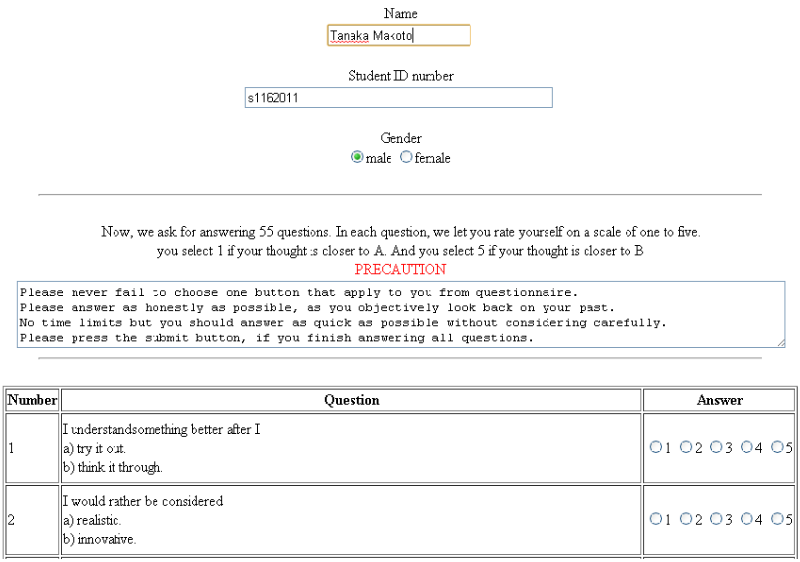


Fig. 5.7 Extended ILS web-based interface

5.3.1.2 Teachers' Module

Teachers have their passwords provided by the system administrator in order to use the system and access the data base. By means of the password they can login to the system and have access to their students' learning preferences data.

An example of the use of our approach by a sample of volunteers is shown in Table 5.4 and Fig. 5.8. Then teachers can access the learning preferences of a single student, a group of students, all students, male students only, and female students only.

The system also has a function to graphically analyze and represent the result for each dimension as it is pictured in Fig. 5.9. The collective result is also displayed graphically as it is illustrated in Fig. 5.10. The system reliability can also be checked through the students' feedback. This reliability is represented graphically and displayed as shown in the Fig. 5.11.

5.3.1.3 Administrator Module

The system administrator may maintain the whole system, create/delete new users (teachers), create or change passwords, and access/maintain the whole data base. The administrator user interface sketched in Fig. 5.12.

Table 5.4 Registered students' information

Id	Name	Gender	ACT_ REF	SNS_ INT	VIS_ VRB	SEQ_ GLB	SOC/ EMO	Feedback
01	Std01	Male	Some Active	Moderate Sensing	Well Balanced	Moderate Sequential	Well Balanced	Right
02	Std02	Male	Some Active	Moderate Sensing	Moderate Visual	Some Sequential	Some Social	Moderately right
03	Std03	Male	Well Balanced	Moderate Sensing	Some Visual	Well Balanced	Well Balanced	Right
04	Std04	Male	Moderate Active	Well Balanced	Some Visual	Some Sequential	Moderate Social	Right
05	Std05	Male	Well Balanced	Well Balanced	Some Visual	Well Balanced	Moderate Social	Moderately right
06	Std06	Male	Well Balanced	Well Balanced	Moderate Visual	Some Sequential	Moderate Social	Right
07	Std07	Female	Well Balanced	Moderate Sensing	Well Balanced	Some Sequential	Some Social	Right
08	Std08	Female	Well Balanced	Well Balanced	Moderate Visual	Some Sequential	Well Balanced	Moderately right
09	Std09	Female	Well Balanced	Well Balanced	Well Balanced	Some Sequential	Some Social	Moderately wrong
10	Std10	Female	Well Balanced	Well Balanced	Well Balanced	Moderate Sequential	Moderate Social	Moderately wrong
11	Std11	Male	Moderate Active	Moderate Sensing	Moderate Visual	Well Balanced	Strong Social	Moderately right
12	Std12	Female	Moderate Active	Some Sensing	Moderate Visual	Moderate Sequential	Moderate Social	Moderately right
13	Std13	Male	Some Active	Some Intuitive	Well Balanced	Well Balanced	Moderate Social	Moderately right
14	Std14	Female	Well Balanced	Moderate Sensing	Some Verbal	Moderate Global	Moderate Social	Right
15	Std15	Female	Well Balanced	Well Balanced	Moderate Visual	Well Balanced	Some Social	Right
16	Std16	Male	Some Active	Well Balanced	Some Visual	Well Balanced	Moderate Social	Moderately wrong
17	Std17	Female	Some Active	Well Balanced	Well Balanced	Well Balanced	Moderate Social	Moderately wrong
18	Std18	Female	Well Balanced	Well Balanced	Well Balanced	Well Balanced	Moderate Social	Right
19	Std19	Male	Well Balanced	Moderate Sensing	Moderate Visual	Moderate Sequent.	Moderate Social	Right
...	...	...	...	...	...	...	...	...

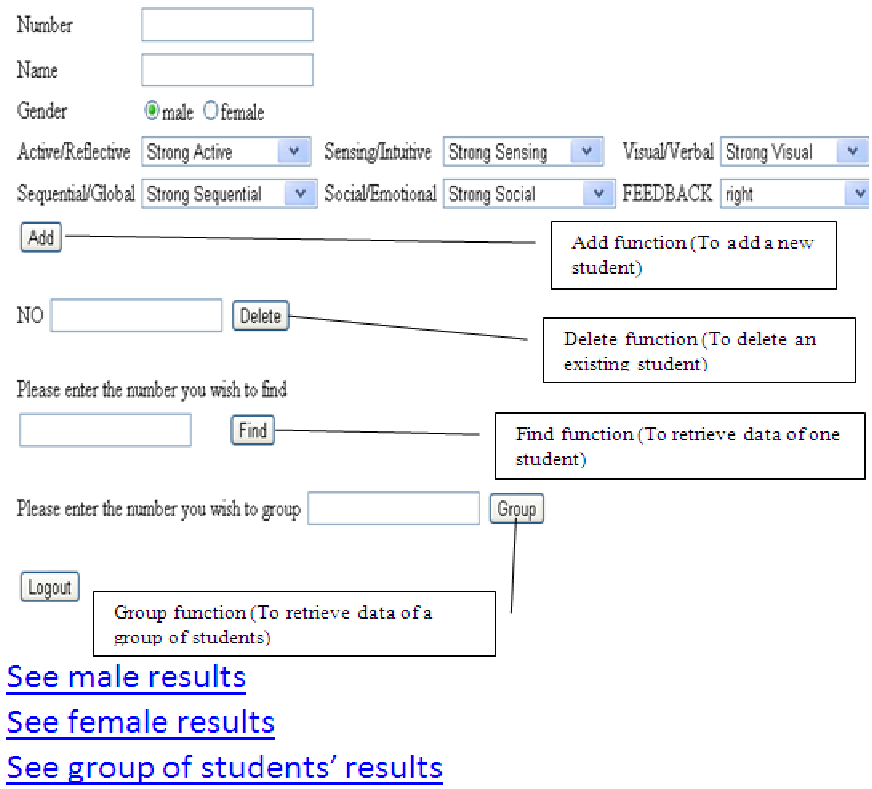


Fig. 5.8 Teacher's module interface

5.3.2 *Support for Intelligent and Adaptive Learning Systems*

An Intelligent and adaptive systems are both model-based systems although they have different purposes to support the learning process. An intelligent tutoring system (ITS) aims to provide the learner-tailored support during a problem solving process, as a human tutor would do. To achieve this, ITS designers apply techniques from the artificial intelligence and implement extensive modeling of the problem-solving process in the specific domain of application (Magnisalis et al. 2011).

On the other hand, the main aim of an adaptive learning system is to adopt some of its key functional characteristics to the learner needs and preferences. For example, content presentation and/or navigation support. Thus an adaptive system operates differently for different learners exactly the way our system behaves.

Our learning style index system can easily be integrated into intelligent and adaptive learning systems as we outline in Sect. 5.5.

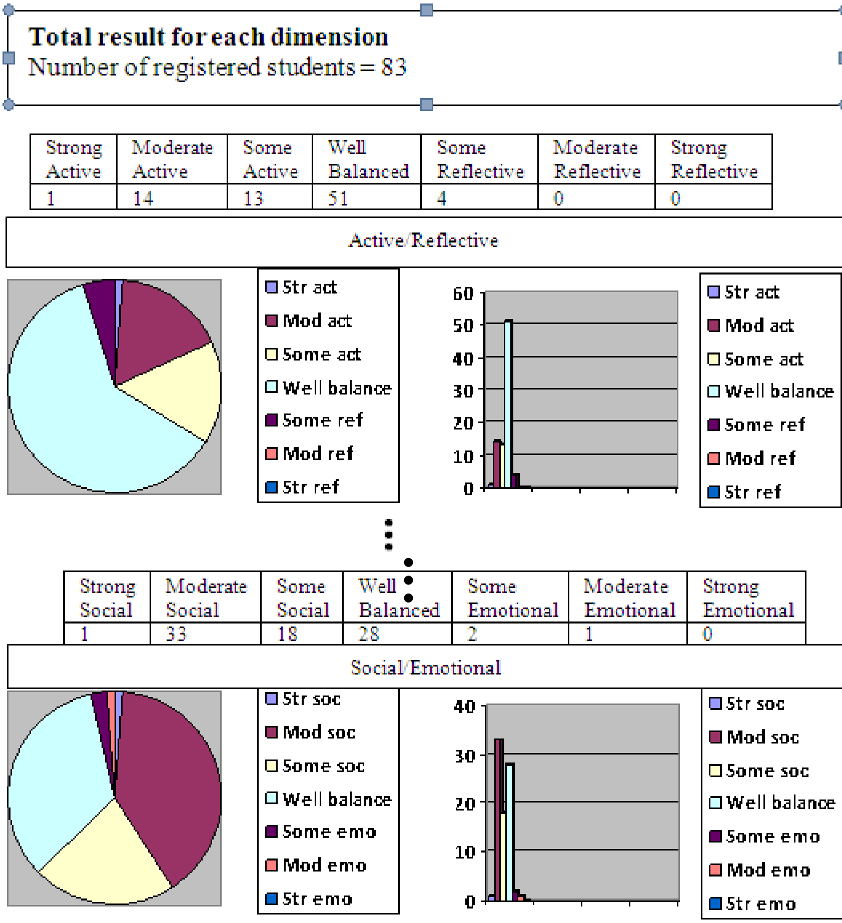


Fig. 5.9 An example of graphical representation of the learning preferences for each dimension

5.4 Application

In the western style education, individual-focused learning is well accepted and the use of LSI and e-learning systems is more efficient. However, in Japan the use of LSI is not so common. It is quite typical in Japanese schools, the classes are given in a traditional lecture-driven style, which means that the difference in learning styles of an individual student has been neglected.

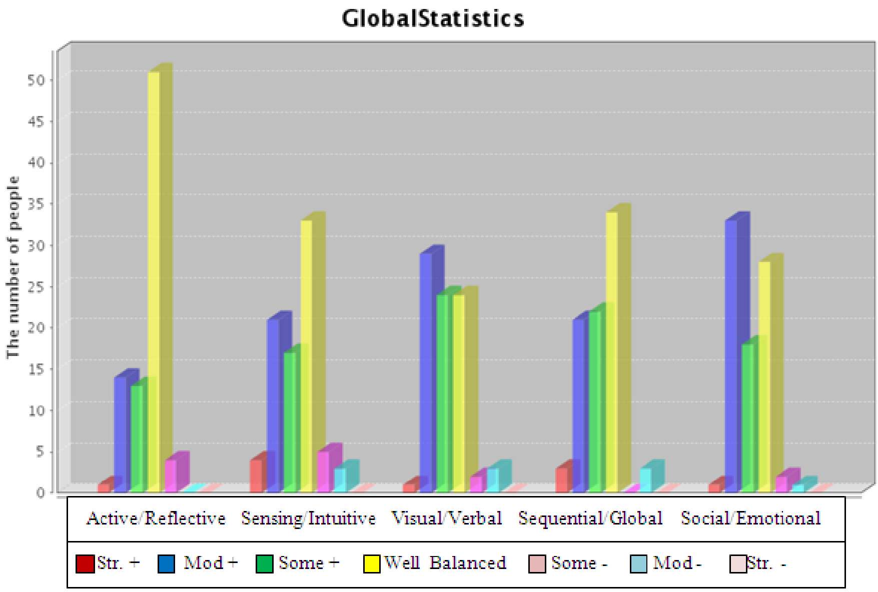


Fig. 5.10 Graphical representation of the collective learning preferences for a group of students

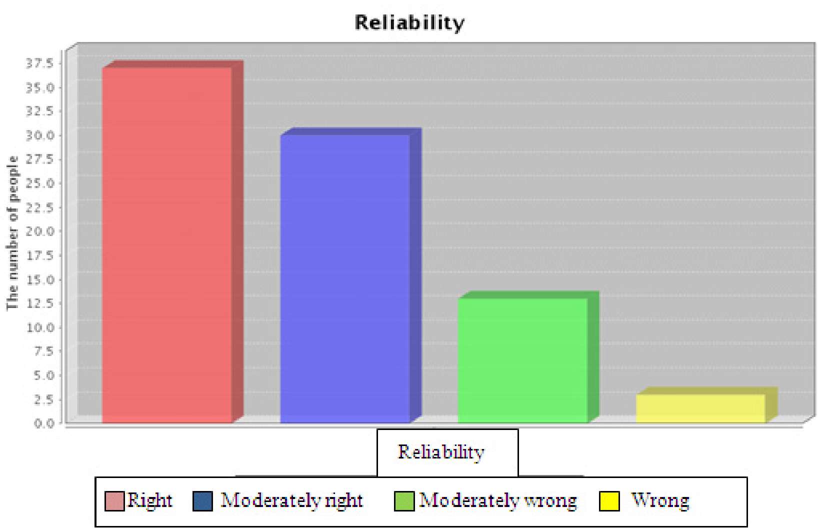


Fig. 5.11 Graphical representation of the system reliability

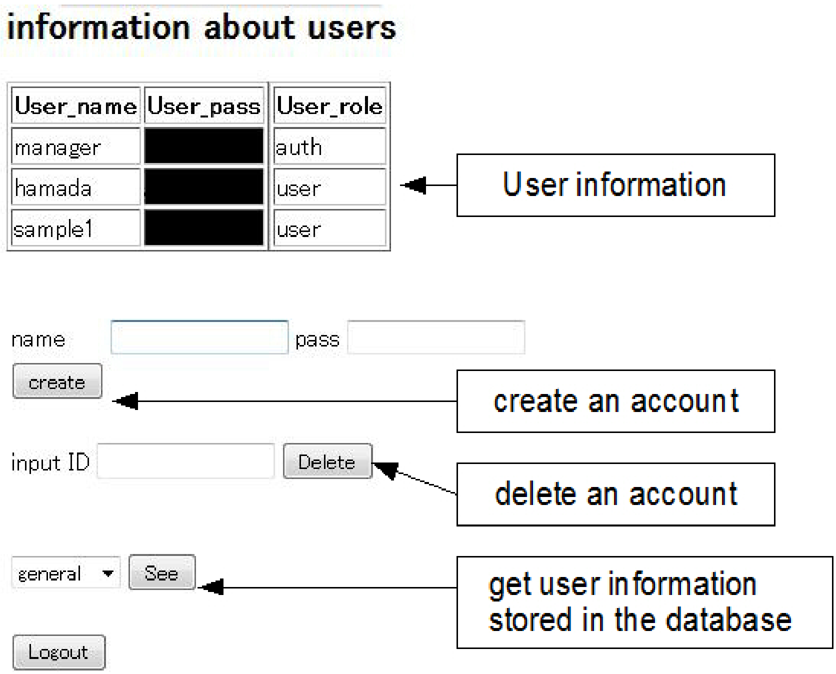


Fig. 5.12 Administrator's module interface

In recent years Japan's educational system has shifted from "collectivity" to "individuality" in line with the advancement of individual-focused learning such as e-learning systems. Under these circumstances, it is more suitable to make the learning process more responsive to individual learners. In this section we introduce the use of our enhanced ELSI for junior high school students.

Our sample consists of 83 students: 26 boys and 57 girls. All of them are second-year junior high school students. The result of the questionnaire is listed in Tables 5.5 to 5.9.

Table 5.5 Active-Reflective learners' distribution

Set	Strong preference for Active	Moderate preference for Active	Some preference for Active	Balance Active-Reflective	Some preference for Reflective	Moderate preference for Reflective	Moderate preference for Reflective
Boys	4% (1)	15% (4)	46% (12)	15% (4)	4% (1)	12% (3)	0
Girls	3% (2)	23% (13)	44% (25)	11% (6)	3% (2)	12% (7)	0
Total	5% (3)	21% (17)	45% (57)	12% (10)	5% (3)	12% (10)	0

Table 5.6 Sensory-Intuitive learners' distribution

Set	Strong preference for Sensory	Moderate preference for Sensory	Some preference for Sensory	Balance Sensory – Intuitive	Some preference for Intuitive	Moderate preference for Intuitive	Moderate preference for Intuitive
Boys	0	12% (5)	4% (1)	42% (1)	19% (5)	19% (5)	4% (1)
Girls	0	9% (5)	9% (5)	58% (3)	15% (9)	9% (5)	0%
Total	0	10% (8)	7% (6)	53% (4)	17% (14)	12% (10)	1% (1)

Table 5.7 Visual-Verbal learners' distribution

Set	Strong preference for Visual	Moderate preference for Visual	Some preference for Visual	Balance Visual - Verbal	Some preference for Verbal	Moderate preference for Verbal	Moderate preference for Verbal
Boys	8%(2)	19% (5)	2% (6)	42% (11)	8% (2)	0%	0
Girls	0%	21% (12)	21% (12)	43 %(24)	12% (7)	3%(2)	0
Total	2%(2)	21% (17)	22% (18)	42% (55)	11% (9)	2%(2)	0

Table 5.8 Sequential-global learners' distribution

Set	Strong preference for Sequential	Moderate preference for Sequential	Some preference for Sequential	Balance Sequential -Global	Some preference for Global	Moderate preference for Global	Moderate preference for Global
Boys	4%(1)	12%(3)	12%(2)	52% (14)	8%(2)	12% (5)	0
Girls	0%	2%(5)	8%(4)	77% (42)	8%(5)	4%(6)	0
Total	1%(1)	6%(5)	8%(7)	71% (56)	7%(5)	7%(9)	0

Table 5.9 Social-emotional learners' distribution

Set	Strong preference for Social	Moderate preference for Social	Some preference for Social	Balance Social – Emotional	Some preference for Emotional	Moderate preference for Emotional	Moderate preference for Emotional
Boys	0	15%(4)	5%(1)	56% (14)	13% (3)	5% (1)	0
Girls	0	7%(4)	11%(6)	47% (42)	24% (14)	16% (9)	0
Total	0	11%(8)	8%(7)	51% (56)	19% (17)	11% (10)	0

5.4.1 Active/Reflective

The responses of boys were normally-distributed, but girls showed some preference for *active*. Because second-year junior high school students were surveyed in this questionnaire, and since females tend to mature earlier than boys, both physically and socially, girls may tend to prefer active learning. Therefore, teachers should try to increase opportunities for group discussion and for experimental and practical lessons. This finding may raise learning efficiency.

5.4.2 Sensory/Intuitive

Both boys and girls display some preference for intuitive learning. 19 percent of boys have a little-to-moderate intuitive preference. Boys are more *intuitive* than girls. Intuitive learners tend to be better at grasping new concepts and are often more comfortable than sensors with abstractions and mathematical formulations. Intuitive learners have more interest in studying science. This may explain why in Japan most of the science and engineering students are boys. Therefore, teachers should try to explain interpretations or theories that link the facts or connections.

5.4.3 Visual/Verbal

Half of the students tend to be visual. We can tell that both boys and girls show a high visual preference. Visual learners remember best what they see. They like to see pictures and diagrams, and they are willing to make concept maps or mind maps. Such kind of graphical and mental representations are the most appropriate ways to learn for students having this preference. In summary, it may be better for junior high school students to assimilate knowledge directly. The students who interpret knowledge are decreasing. Therefore, teachers should try to use visual material in class, which may also raise learning efficiency for this group.

5.4.4 Sequential/Global

Neither boys nor girls show a bias towards sequential or global orientation. Sequential and global distribution are fairly distributed. Teachers should try to be concise regarding two dimension's poles because most of the students are balanced on sequential and global. At the beginning of each lesson, teachers should explain the outline of the topic in logical order and how it relates to the real life subjects and facts.

5.4.5 Social/Emotional

The respondents tends to concentrate centrally on this dimension. However, a greater percentage of boys display a preference for social learning than girls. On the other hand girls indicate a greater preference for emotional learning than boys. This result is perhaps not surprising if we consider the possible gender-biasing effects of Japanese culture on the social behavior of people.

In conclusion, our survey revealed that most of the students in the sample tend to have "well balanced" learning preferences. However a considerable number of them tend to have "visual" learning preferences.

5.5 Integration into Intelligent and Adaptive E-Learning System

Compared to traditional learning systems, e-learning (Advanced Distance Learning Group, ADL 2007) provides a more comfortable learning environment, where learners can learn at their convenience. E-learning systems are widely used and rapidly increasing.

Hamada 2008 built an e-learning system for automata theory and theory of computation based on Java2D technology (Sun Microsystems 2006). Such a system is illustrated in Fig. 5.13. Hamada's e-learning system is an intelligent and adaptive learning system that embraces the next components:

- Animated (movie-like) welcome component,
- Hypertext introduction to the theory of computation topics,
- Finite state machine (FSM) simulator,
- Turing machine (TM) simulator,
- Self-assessment component,
- Chatting component for supporting online collaborative learning,
- Other components showing visual automata examples such as a video player, rice cooker, and tennis game.

Novice automata learners find it difficult to grasp these comprehensive materials that were designed to meet all kinds of learning preferences. Learners do not know where they should start. In order to overcome such an issue, we extend Hamada's e-learning system by adding a new component for learning style. This new component, sketched in Fig. 5.14, enables the user to find his/her learning preferences and hence to choose suitable components from the rich automata e-learning system.

The integration of our enhanced learning-style system into Hamada's automata e-learning system requires getting access to the source code. Fortunately, since both systems are written in Java, there was no compatibility problem in the integration process.

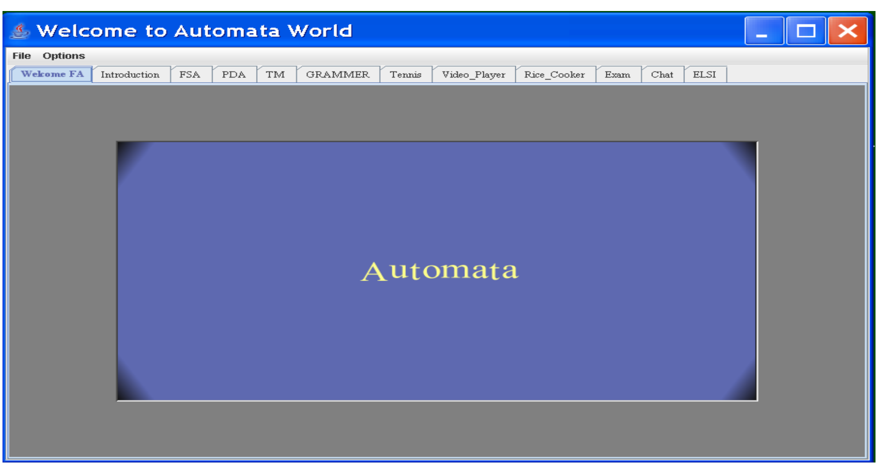


Fig. 5.13 Automata e-learning system interface

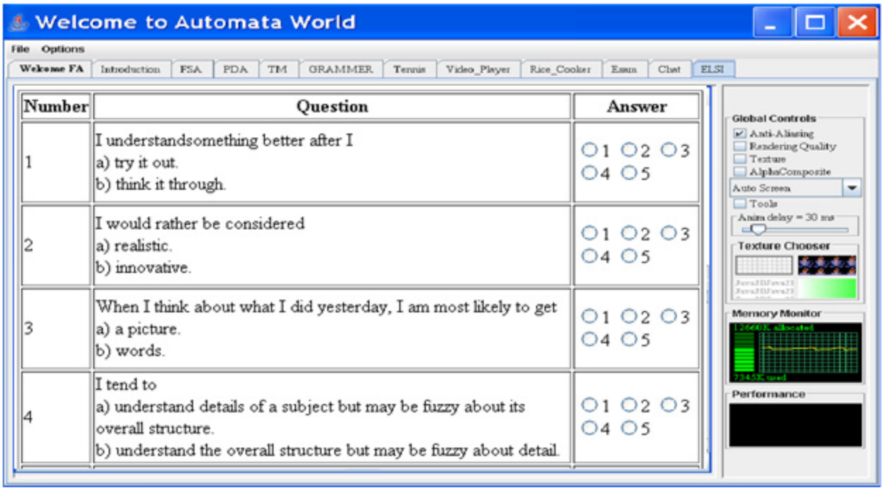


Fig. 5.14 EILS integrated into Hamada's automata e-learning system

5.5.1 *Learning Activities*

When using the EILS component of the Automata learning system, the user gets a set of recommendations to start studying automata based on her/his learning preferences. For example, a visual learner is recommended to select the following set of activities, which learners can consider when using the environment:

1. Start using the environment by playing with the visual examples. This does not need any special knowledge or background regarding the topics. It also will attract the learners' attention to the relevance of the topics. Learners' attention and topic relevance are the basics to Keller's ARCS motivation model (Keller 1987),
2. Take the first simple general test. By answering the easy and general questions in this test the learner gains familiarity and self-confidence which is an important factor for learners' motivation in ARCS motivational model (Keller 1987). At this stage learners are ready to start reading the theoretical concepts in the topics object,
3. Navigating the concepts in the topics object provides the learners with the necessary theoretical background for the subject,
4. Start using the FSM and the TM simulators. Switching between reading the topics and using the simulators are recommended. After reading a certain topic, the learner can switch to the simulator and try to build a model for that topic and test the model with different inputs. This can help in deepening the learners' knowledge and can enhance the learning process,
5. While reading the topics and using the simulators, learners are recommended to try the corresponding test (in the test object) for self-assessment and to gain more confidence about their learning progress,

6. At any stage of the learning process, on-line learners can chat with each other through the chatting object. This enables learners to exchange ideas and help each other to understand the topics and answer the test questions in a collaborative way.

The environment objects and the workflow of the learning activities for visual learners are shown in Fig. 5.15. Whereas, reflective learners get a different set of recommended activities as the following:

1. Start by navigating the concepts in the topics object. This will provide the learners with the necessary theoretical background for the subject,
2. Try the corresponding tests starting from test number 1,
3. Play with the visual examples.
4. Use the FSM and the TM simulators. Switching between reading the topics and using the simulators are recommended,
5. At any stage of the learning process, on-line learners chat with each other by the chatting object. This enables learners to exchange ideas, help each other to understand the topics and answer the test questions in a collaborative way.

The environment objects and the workflow of the learning activities for reflective learners are shown in Fig. 5.16.

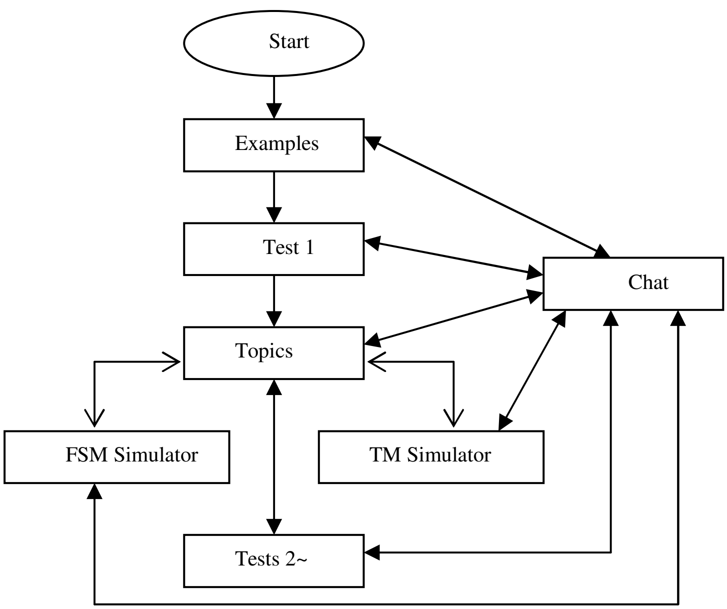


Fig. 5.15 Workflow of learning activities for visual learners

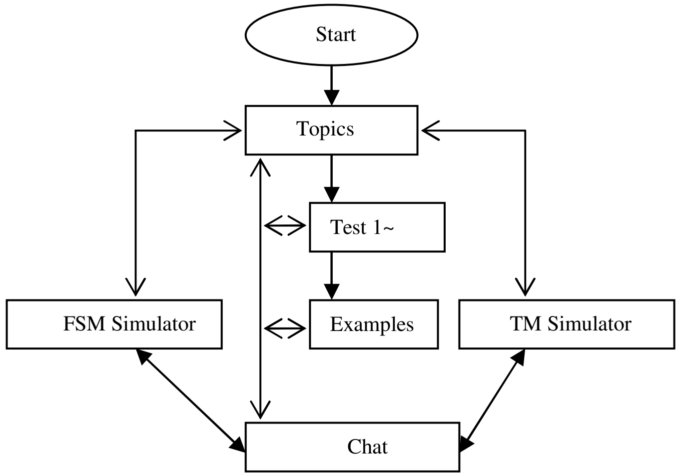


Fig. 5.16 Workflow of learning activities for reflective learners

5.6 Conclusion

In this research, we developed an enhanced version of a LSI that can be integrated into intelligent and adaptive learning systems. We implemented our model in a way that allows learners to easily check their learning preferences.

Moreover, teachers can have a wider perspective on their students' learning preferences. To this extent, our implementation utilizes several useful tools such as: web-based interface, java applets, Apache web server, Tomcat application server, MySQL database, and JDBC connector.

We tested our system on a sample of 83 junior high school students. We inferred their learning preferences as individuals and as groups. Then we analyzed the result and reported our recommendations to their teachers who appreciated the work. However we have not carried out a follow up study of the recommendations. This issue will be considered in future work.

To show the flexibility and usefulness of our implemented system, we integrated it into an intelligent and adaptive e-learning system that is based on Java2D technology and contains an intensive set of learning materials to support all kind of learners. Thus, learners with different preferences will get different sets of learning activities. For example, active learners will be recommended to use relevant materials that match their preferences.

With this integration the automata e-learning system should be more effective since learners can more easily explore and understand the rich set of materials in the system. However this is just a starting point, and a follow-up and evaluation of the integration are necessary. This is what we intend to investigate in our future research.

References

ADL. Website (2007), `http://www.adlnet.gov` (acessed February 21, 2011)

ASF. Website (2008), `http://www.apache.org` (acessed December 01, 2010)

Felder, R., Silverman, L.: Learning and teaching styles in engineering education. Engineering Education 78(7), 674–681 (1988)

Hamada, M.: An integrated virtual environment for active and collaborative e-learning in theory of computation. IEEE Transactions on Learning Technologies 1(2), 1–14 (2008)

Herrmann, N.: The Creative Brain. Brain Book, Lake Lure (1990)

Keller, J.: Development and use of the ARCS model of motivational design. Journal of Instructional Development 10(3), 2–10 (1987)

Kolb, D.: Experiential learning: experience as the source of learning and development. Prentice-Hall, Englewood Cliffs (1984)

Kort, B., Reilly, R., Picard, R.W.: An affective model of interplay between emotions and learning: reengineering educational pedagogy- building a learning companion. In: Proceedings of ICALT, pp. 43–46. IEEE Press, New York (2001)

Kumiko, F., Mari, M.: Cloninger's temperament dimensions, emotional experiences and emotional regulation. Yamagata Univ. Educ. Sci. 14(4), 387–397 (2009) (in Japanese)

Magnisalis, I., Demetriadis, S., Karakostas, A.: Adaptive and intelligent systems for collaborative learning support: a review of the field. IEEE Transactions on Learning Technologies 4(1), 5–20 (2011)

Silvia, R.V., Sabine, G., Kinshuk, T.L.: Analysis of Felder-Silverman index of learning styles by a data-driven statistical approach. In: Proceedings of ISM, pp. 959–964. IEEE Press, New York (2006)

Silvia, R.V., Sabine, G., Kinshuk, T.L.: Investigating relationships within the index of learning styles; a data driven approach. Journal of Interactive Technology and Smart Education 4(2), 7–18 (2007)

Soloman, B., Felder, R.: Index of learning style questionnaire (2009), `http://www.engr.ncsu.edu/learningstyle/ilsweb.html` (acessed January 20, 2010)

Sun Microsystems. Java2D (2006), `http://www.sun.com` (acessed March 11, 2010)

Thomas, A.L., Sang, H.L., Wise, J., Richard, F.: A Psychometric study of the index of learning styles. Journal of Engineering Education 96(4), 309–319 (2007)

Tomcat. Website (2010), `http://tomcat.apche.org` (accessed December 01, 2010)

Abbreviations

ADL	Advanced Distance Learning
A/R	Active/Reflective
ARCS	Attention – Relevance – Confidence – Satisfaction model
ASF	Apache Software Foundation
DIM	Dimension
ELSI	Enhanced Learning Style Index
FSM	Finite State Machine
ILS	Index of Learning Style (see also LSI)
ITS	Intelligent Tutoring System
Java2D	Java 2 Dimension

JDBC	Java DataBase Connect
LSI	Learning Style Index (see also ILS)
MySql	My SQL (see SQL)
PC	Personal Computer
SEL	Social Emotional Learning
S/G	Sequential/Global
S/I	Sensory/Intuitive
SQL	Structured Query Language
TCI	Temperament and Character Inventory
TM	Turing Machine
V/V	Visual/Verbal

Chapter 6
GRAPPLE[1]: Learning Management Systems Meet Adaptive Learning Environments

Paul De Bra[1], David Smits[1], Kees van der Sluijs[1], Alexandra I. Cristea[2], Jonathan Foss[2], Christian Glahn[3], and Christina M. Steiner[4]

[1] Eindhoven University of Technology
Den Dolech 2, Eindhoven, The Netherlands
debra@win.tue.nl, {d.smits,k.a.m.sluijs}@tue.nl
[2] University of Warwick
Coventry CV4 7AL, United Kingdom
{acristea,jonny}@dcs.warwick.ac.uk
[3] Open University in The Netherlands
Valkenburgerweg 177, Heerlen, The Netherlands
Christian.Glahn@ou.nl
[4] Graz University of Technology
Rechbauerstraße 12, Graz, Austria
christina.steiner@tugraz.at

Abstract. Learning Management Systems (LMSs) are used in many (educational) institutes to manage the learning *process*. Adaptive Learning Environments (ALEs) offer support for the *learning* process through adaptive guidance and perhaps also personalized learning material (content). GRAPPLE offers a new infrastructure that brings both together. This is done through single sign-on, a common User Model Framework and an (asynchronous) event bus that coordinates the communication between the other components. Authors can create structured course material and define the adaptation through a graphical interface, and a flexible and very extensible adaptation engine offers almost any type of presentation and adaptation an author might want. This chapter reports on early experience with the GRAPPLE environment, for teaching and for learning.

6.1 Introduction

In the past 15 years two complementary technologies have been introduced in the area of learning: Learning Management Systems (LMS) and Adaptive Learning Environments (ALE).

---

[1] GRAPPLE stands for "Generic Responsive Adaptive Personalized Learning Environment and is the name of an EU FP7 STREP project.

A. Peña-Ayala (Ed.): Intelligent and Adaptive ELS, SIST 17, pp. 133–160.
springerlink.com

An LMS (such as the popular Blackboard[2], Sakai[3], Moodle[4], etc.) supports the learning *process* and administration. Teachers can create courses that students can enroll in. An LMS can support creating and serving tests, grading assignments, publishing course material, and it can allow communication through chat rooms, discussion forums, etc. An ALE supports the *learning* itself, by means of personalized access to course material. Whereas an LMS is most beneficial for the institute (university, company) an ALE is most beneficial for the learner.

The GRAPPLE project (De Bra et al. 2010) brings the world of LMSs and ALEs together in order to offer a life-long learning solution. One of the main issues in adaptive learning is that the system needs a rich representation of the learner. GRAPPLE introduces a common, shared and distributed framework GUMF (GRAPPLE User Modeling Framework) (Abel et al. 2010) that allows different instantiations of ALEs and LMSs to consult and update information about learners. The overall GRAPPLE infrastructure is shown in Fig. 6.1 below. We briefly describe it here and provide more details in Sect. 6.4.

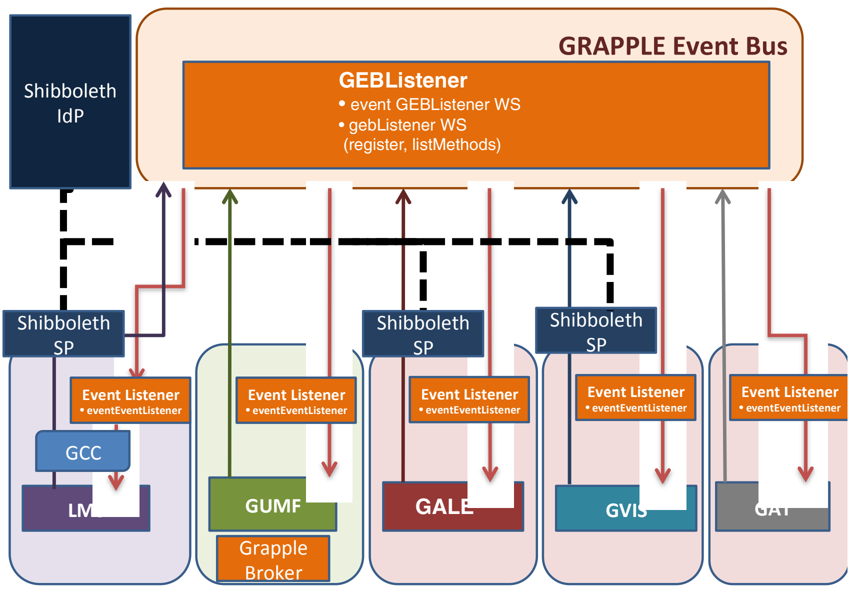


Fig. 6.1 The overall GRAPPLE architecture

2 www.blackboard.com/
3 sakaiproject.org/
4 moodle.org/

In addition to this user information GALE maintains a detailed User Model (UM) used for fine-grained adaptation within a course. GALE may perform adaptation within a course based on exactly which pages the learner has studied in that course, and periodically informs GUMF about the learner's progress, probably at the chapter or larger topic level, for use in the adaptation of other courses for instance. The adaptation performed by GALE is described in detail in Sect. 6.2. In order to demonstrate a large scale merger between LMS and ALE, within the GRAPPLE project the Open Source LMSs Claroline[5], Moodle and Sakai and the commercial systems Clix[6] and learneXact[7] have been integrated into the GRAPPLE infrastructure. Section 6.4 shows some details of (configuring) this integration. Part of the difficulty of integrating LMSs with GALE is that information about users is described in different terms and scales in each LMS and in GALE. In order for all components to be able to use each others' user information the LMSs are extended with a GRAPPLE Conversion Component (GCC) to store information in GUMF in an agreed upon way.

Teachers use the GRAPPLE Authoring Tool (GAT) (Hendrix et al. 2008, Hendrix et al. 2009, Hendrix and Cristea 2009) to create the conceptual and pedagogical structures of a course, and to associate content with the concepts. The adaptation is based on standard pedagogical relationship types (PRTs) (Albert et al. 2009) like *prerequisites*; however in GAT an author creates new PRTs and associate arbitrary adaptive behavior with these PRTs. GAT is described in Sect. 6.3.

Teachers and learners can use the visualization tool GVIZ to view progress reports. We do not describe GVIZ in detail in this chapter.

The five basic components of GRAPPLE: LMS, GUMF, GALE, GVIZ and GAT, shown in Fig. 6.1., are connected in two ways: a) All components use a common asynchronous event bus (GEB) to exchange information. The use of such an event bus means that all components need to understand communication only with that event bus, not with all the other components. b) The three components used directly by the learners: LMS, GALE, and GVIZ, are all connected to the single sign-on facility Shibboleth[8] (not developed within the GRAPPLE project). A learner who is logged in on the LMS of his institute (university or company) can directly "click through" to an in-line presentation of the adaptive course text and can receive in-line visualizations of his progress. The combined LMS, GALE and GVIZ thus present themselves as a single learning environment, possibly in a single browser window. (The actual presentation form depends on the LMS used and possibly on configuration options within that LMS.)

In this chapter we show how *prerequisites* and *content adaptation* (conditionally included fragments) can be combined to allow learners to study an on-line course text in almost any desired order without encountering learning material they are not ready to understand.

[5] www.claroline.net/
[6] www.im-c.de/
[7] www.giuntilabs.com/
[8] shibboleth.internet2.edu/

In Sect. 6.5 we show a number of example courses that were developed using the GRAPPLE tools and reflect on our experience with creating adaptive learning material as well as with studying by means of adaptive course material. The example courses created using the first set of PRTs delivered with GAT all follow a form of *informal learning* which we call *learning through information exploration*. Creating such courses is easy as it requires no technical skills and is done entirely through a graphical interface. This ease of use does have the drawback that it does not make use of the full potential of GRAPPLE.

The ability to create new PRTs and to associate arbitrary adaptive behavior with each type not only allows different levels of guidance but also allows authors to add different levels of intelligence and more complex adaptation strategies to an on-line course. Adaptation can be based on learning styles (Stash et al. 2006, Stash et al. 2008) for instance to offer information in different media types or to advise a different learning order to learners with activist/reflector or global/sequential learning styles. Because learner information is stored (in GUMF) to be reused in future courses the adaptation can get better as a learner is taking more GRAPPLE-based courses (even at different institutes, using a different LMS). The chapter illustrates some examples of such applications; however we wish to stress that GRAPPLE is a very general/generic environment that can realize many more and different intelligence and adaptation levels than any series of examples can show.

6.2 Adaptive Learning Methods and Techniques

Adaptation in learning is often considered a problem of performing adaptive *course sequencing*, see e.g. (Gutierrez-Santos et al. 2008), and is supported by standards such as IMS Learning Design[9]. In the GRAPPLE project we approach the issue of adaptation from the viewpoint of *adaptive hypermedia*. In hypermedia the end-user (or learner) has a lot of navigational freedom, to browse through course material by following arbitrary links. Adaptation guides the learner without creating sequences or any other form of specific and/or restrictive workflow, as was common in Intelligent Tutoring Systems (ITS) that dominated the Technology-Enhanced Learning scene until about 15 years ago. Brusilovsky (Brusilovsky 1996, Brusilovsky 2001) defined and Knutov (Knutov et al. 2009) updated/refined a taxonomy of adaptive methods and techniques, shown in Fig. 6.2 below. We provide this taxonomy mainly for reference, to show the plethora of different ideas adaptive hypermedia researchers have come up with over roughly the past 15 years. We will explain some techniques below, but we do not have space to explain all of them and in addition to explain how they can be realized using GRAPPLE technology.

[9] www.imsglobal.org/learningdesign/

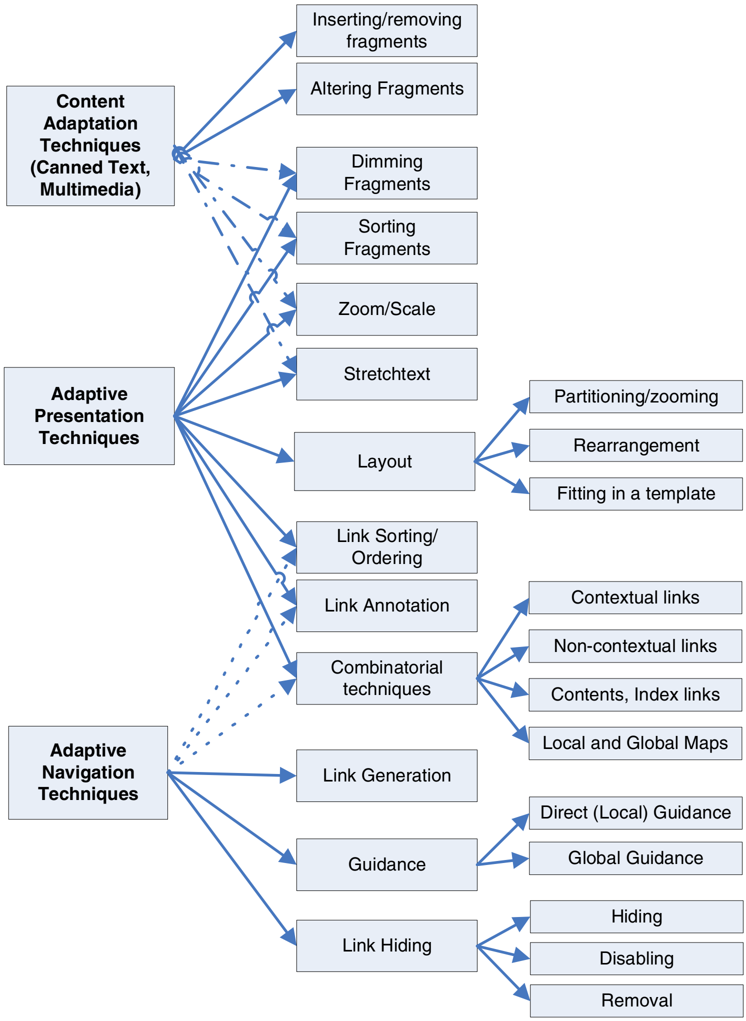


Fig. 6.2 Adaptive methods and techniques (Knutov et al. 2009)

We describe the techniques mostly in a "generic" way here. Section 6.5 has a concrete course example (the "Milkyway" course, about celestial objects, mainly those of our solar system) in which the adaptation techniques are used and can be seen.

Designing an ALE to support this wide variety of adaptive methods and techniques to provide a truly personalized learning experience is a near-impossible task. In GRAPPLE this issue was tackled by creating GAT (the GRAPPLE Authoring Tool) and GALE (the GRAPPLE Adaptive Learning Environment).

GAT allows adaptation to be defined at a *conceptual* and *pedagogical* level, using graphical tools, and is explained in Sect. 6.3. GALE is concerned with adaptively selecting and presenting *resources* (files, or pages) and has been designed as a flexible and extensible adaptation engine to which "missing" functionality (techniques from Fig. 6.2 not yet supported by GALE) can easily be added.

GALE follows the taxonomy of Fig. 6.2 by offering *content* adaptation, *presentation* adaptation and *link* adaptation. To save space we will only describe the content and link adaptation (and not the possibilities to have adaptive layout, style sheets or to perform device adaptation).

- GALE is preconfigured to provide adaptation to XML (or XHTML) resources (or files). For a number of XML tags GALE offers a *module* that performs adaptation to the tag or corresponding element. Developers can easily add modules and associate them with specific tags. Some of the default tags and corresponding modules are:
  - The <if> tag (and corresponding <then> and <else> tags) is used to conditionally include a fragment of content. The condition is an expression over the *Domain Model* (DM) and UM, explained in Sect. 6.3. This realizes the *insertion/removal of fragments* technique, and to some extent also the *altering fragments* technique. <if> can be used to present a short explanation of a term when according to UM the user does not yet know that term. But <if> can also be used to present something depending on DM: in the Milkyway example we will see that the whole paragraph about the moons of a planet only appears for planets that actually have moons according to their definition in DM,
  - An alternative way to conditionally insert fragments is by using the <object> tag, which (in standard XHTML) normally inserts a file with a fixed given name, but in GALE inserts whatever resource is (adaptively) associated with a *concept* from DM,
  - The <for> tag is used to generate a sorted list of fragments, each corresponding to an item in a list of concepts. We show an example in Sect. 6.5 (generating a list of moons of a planet). This realizes the *sorting fragments* technique,
  - There are some additional tags (and modules) to insert DM or UM information in a presentation directly, for instance the <variable> tag to insert the value of an expression in the page and <attr-variable> to insert the value of an expression into an XHTML (or GALE) tag. For instance, this can be used to show the learner's name and email address in a header. The <count> tag is used to show the learner's progress: how many pages the user has read and/or how may pages still have to be studied. Again, Sect. 6.5 shows this generated information in the header of the Milkyway example.

- The *adaptive navigation techniques* offered by GALE are:
  - The <a> tag (used for link anchors) no longer refers to a resource (file) but to a concept. In Sect. 6.3 we describe how, similarly to the conditions of an <if> tag, we can use expressions to adaptively choose between a number of possible link destinations. Different learners who see and click on the same link may thus jump to different pages depending on what they studied before. This implements the *link generation* techniques (it can be *URL adaptation* and *destination adaptation*),
  - By combining the <a> tag with style sheets the presentation of the link anchors can be adapted (GALE can do this using colors and icons). This realizes the *link annotation* technique, and it can also realize the *link hiding* technique. Although GALE can use any number of arbitrary colors we typically use the colors blue, purple and black to indicate recommended unvisited, recommended visited and non-recommended links,
  - GALE has a predefined "next" view (the <view> tag is used to insert a view) which adaptively chooses the most suitable concept to study next. It is thus possible to *suggest* a learning sequence (*without enforcing it*). This realizes *direct (local) guidance*,
  - GALE also has views for generating navigation menus, offering different forms of *fisheye views* or *zooming* into the overall conceptual structure of a course. The Milkyway example in Sect. 6.5 shows an automatically generated *accordion menu* of chapters and sections of the course.

Clearly, when creating adaptive courses one has to be careful to choose wisely which of the many adaptation techniques (of Fig. 6.2) to use. GALE offers many possibilities (and can be extended to offer even more, for instance to use a combination of tags and style sheets to perform *dimming fragments*, *stretchtext* and *zooming/scaling*). Evidently the *less is more* principle applies: using a few techniques in a consistent way is better than creating a showcase of all possibilities of adaptivity in a single course.

The real "intelligence" for the adaptive behavior of GALE comes from adaptation rules generated by the authoring tool described in the next section. Just like in *active databases* GALE uses *event-condition-action rules* (ECA rules) to generate user model updates and adaptation behavior from end-user-generated events such as accessing a concept or performing a (multiple-choice) test.

A rule that generates a user model update may trigger other rules that generate more updates, etc. The use of ECA rules allows for arbitrarily complex intelligence to be defined but in general makes this intelligence very hard to define (and also to avoid inadvertedly generating infinite loops or unpredictable results). The use of *template* code in the authoring tool keeps this manageable for a non-technical author.

6.3 Creating Adaptive Course Material for GRAPPLE

In GRAPPLE we mainly consider *authoring* as an activity at the *conceptual* level. Clearly, pages (or other forms of learning material) have to be written, however, as GRAPPLE mainly uses standard Web technology (access to resources through HTTP, and pages written in XHTML with possibly small extensions) this part of the authoring process is left out of this chapter, even though it takes up most of the effort (time) of authoring. Section 6.2 described small extensions to XHTML that allow adaptation within a page. In this section we only describe the additional effort needed to make a course *adaptive* at the more global level.

The GRAPPLE Authoring Tool (GAT) consists of three parts, two of which are used by every course author and one only by very advanced adaptivity designers.

- The *Domain Tool* is used to define the *concepts* of a subject domain, their associated *resources* and the *relationships* between these concepts. This represents an initial content structure for a certain topic. Roughly speaking one can think of a *Domain Model* (DM) as a (part of an) ontology,
- The *Course Tool* is used to define the *adaptive behavior* of a course by means of *Pedagogical Relationships* chosen from a set of available, predefined *Pedagogical Relationship Types* (PRTs). Additionally, the Course Tool also assists in choosing the appropriate domain concepts from the relevant domains defined previously, with the Domain Tool. Note that a course can be composed from several domains – just as in traditional classroom teaching a course can be composed from several books and other sources. The resulting *Course Model* is also called *Conceptual Adaptation Model* (CAM) for historical reasons[10]. It defines how *knowledge* is acquired by studying concepts, and how *suitability* of concepts depends on the *knowledge* of other concepts through different relationships, including *prerequisite* relationships,
- The advanced *Pedagogical Relationship Type Tool* is used to define PRTs (such as *prerequisites*) and their associated adaptive behavior. PRTs are reusable, independent of the course. Due to the lack of space we will only describe the main parts of the PRT definition (and not the details of the tool interface).

The Domain tool and the Course Tool reflect the GRAPPLE principles for authoring tools: they are based on conceptual structures (Hendrix et al. 2008) of hypermedia, graph-like nature (thus, non-hierarchical), which allow for various links between the objects to be represented. Moreover, these tools allow for visualizations which support these types of structures (Hendrix and Cristea 2009), thus moving away from previous text-based representational paradigms. Additionally, the tools internally use Semantic Web-based representations, such as unique URIs for all components, and XML-based export formats for all tools.

---

[10] The term CAM is use in some references (Hendrix et al, 2009, Hendrix and Cristea, 2008, Hendrix and Cristea, 2009). This chapter only uses the term *Course Model*.

6.3.1 The Domain Tool

Fig. 6.3 shows a screenshot of the Domain Tool. It depicts the Milkyway domain (Ploum 2009), created by a Masters student at the Eindhoven University of Technology (TU/e) and explained more in detail in Sect. 6.5.

The Domain Tool uses a graphical presentation (and editor) for *concepts* and named or typed *domain relationships*. For each concept of a Domain (or DM) the tool is used to define the following information:

- Name: each concept has a unique name within the DM; both the name of the course and the concept are used in the creation of a URL to access the concept through a link,
- Description: this is a reminder for the author what the concept is about but is not used in any way when the learner is interacting with the course,
- Resources: each concept can be associated with a number of resources; obviously only one resource can be presented (as a page) to the learner, and one way in which a resource can be selected is by associating a Boolean *expression* (over DM and UM) with the resource: the first resource with an expression that evaluates to true is retrieved, adapted and presented. Note that concepts need not be associated with a resource: you can have an abstract concept to represent only aggregated knowledge from smaller concepts (but not having a presentation of its own) and you can also have concepts that are used to gather a knowledge level obtained by taking a test in the LMS (the LMS passes its grade book items to GALE through GUMF),
- Properties: each concept can have arbitrarily many properties, each having a name and a value (and a description which is not used by the system); typical properties are a *title* to be used to refer to the concept in a navigation menu, a *type* used by some views, for instance to show links to or to count a number of *page* concepts, and *order* to define how to sort a list of concepts. In the Milkyway example we have also used additional properties, like *image* and *info*, which are URLs of objects to be included in the presentation; in this way, complicated adaptation can be achieved without the author needing to specify complex adaptation. All they need to do is to add the properties corresponding to already predefined adaptation,
- Relationships: concepts are typically arranged in a hierarchy, by linking them through *parent* relationships. This corresponds to the classical way teachers organize their material. However, in the GRAPPLE Domain Tool, in addition to that, concepts can participate in other relationships, as illustrated in the Milkyway example, which is using specifically designed relations called *isMoonOf*, *isPlanetOf* or more generally, *rotatesAround*; all these relationships can be used in the course adaptation (and in fact are used in our example).

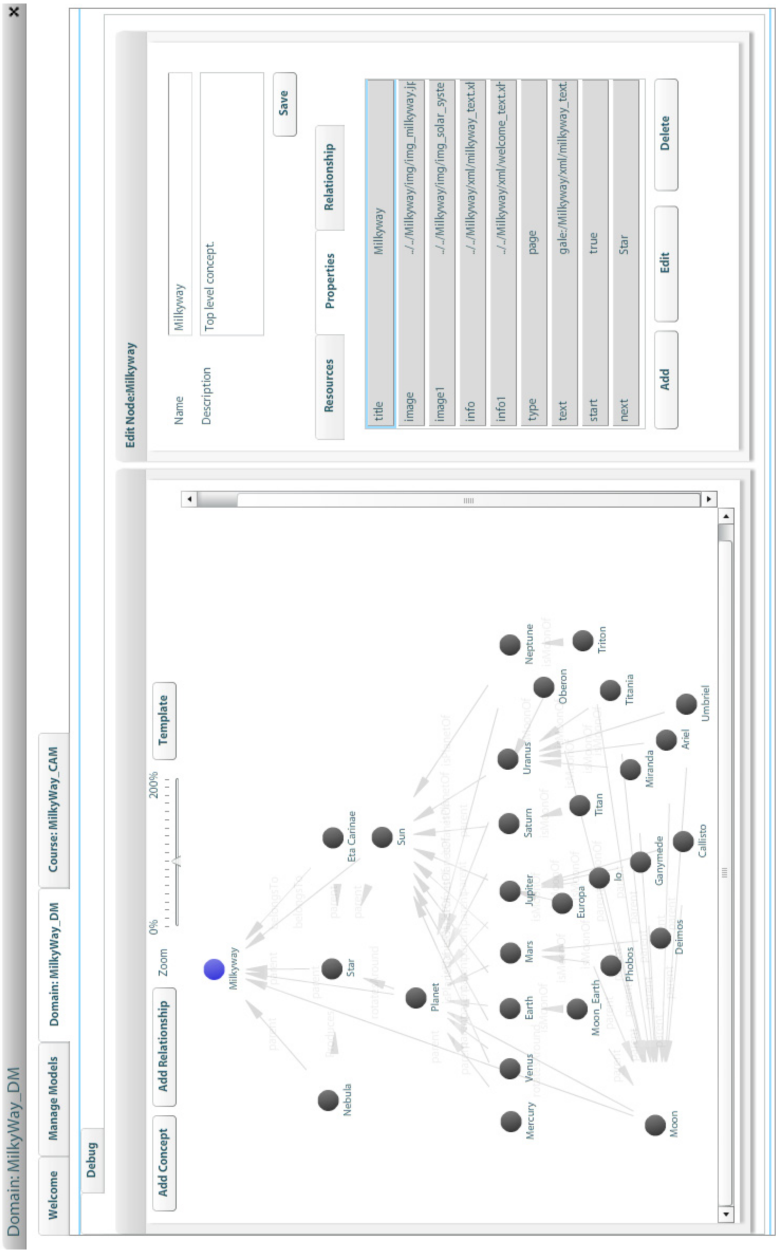


Fig. 6.3 The Domain Tool (showing the Milkyway course)

The basic idea of the DM is to only represent subject domain information, which is independent of the teaching/learning process. From the description above (and also illustrated by Fig. 6.3) it is clear that it is tempting for authors to already mix adaptation hints into the DM. Expressions for selecting resources, and the *order* attribute to guide menu construction are "violations" of the strict separation of the DM and adaptation. Such a violation can be solved by introducing a separate layer of pedagogical metadata, such as advocated by the LAOS framework and implemented by us in previous research (Cristea and Mooij 2003).

6.3.2 The Course Tool

To teach or learn about the subject domain of a course the Course Tool helps define a *pedagogical* structure, based on PRTs. This structure expresses how studying a course leads to user model (UM) updates, and how the UM state tells the adaptation engine how to provide guidance. In other words, the Course Tool combines information on *what* material is used (as inserted from the Domain) with *how* this material is to be adaptively presented (as inserted via Pedagogical Relationship Types). Fig. 6.4 shows the Course Model for the example course Milkyway.

The Milkyway example has been used several times in workshops where potential authors of adaptation were asked to define a pedagogical structure based on the *learning through information exploration* paradigm. The participants had to choose a start concept (which would be the same for every learner) and a then devise a structure of *prerequisite relationships* to be used to guide the learner. During these workshops we found that whereas the DM that was given to the participants contained binary domain relationships between (single) concepts the participants felt the need to define prerequisite relationships with multiple concepts in the *source* or the *target*. For instance, a planet would be a prerequisite for all of its moons. This was at times difficult for the workshop participants to draw on paper, but in the Course Tool this is facilitated through the use of *sockets*.

The Course Tool contains instances of different PRTs which previous experience and system usage showed to be the most frequently needed pedagogical relations. This basic set contains PRTs of various simple kinds, some of which have one socket containing concepts and others having two sockets, called *source* and *target*. In general, PRTs can have any number of sockets as desired by the author. However, such complex PRTs can only be defined by very advanced authors – although once created they can be reused by everyone. Most basic PRTs[11] are listed below, together with their functionality, as available to all authors:

[11] This is not an exhaustive list. Some additional PRTs already exist: *G-Hide*, *G-Unhide*, *G-Visit*, *G-Quiz*, *G-Knowledge-Propagation*, and more will be added in the future as the need arises.

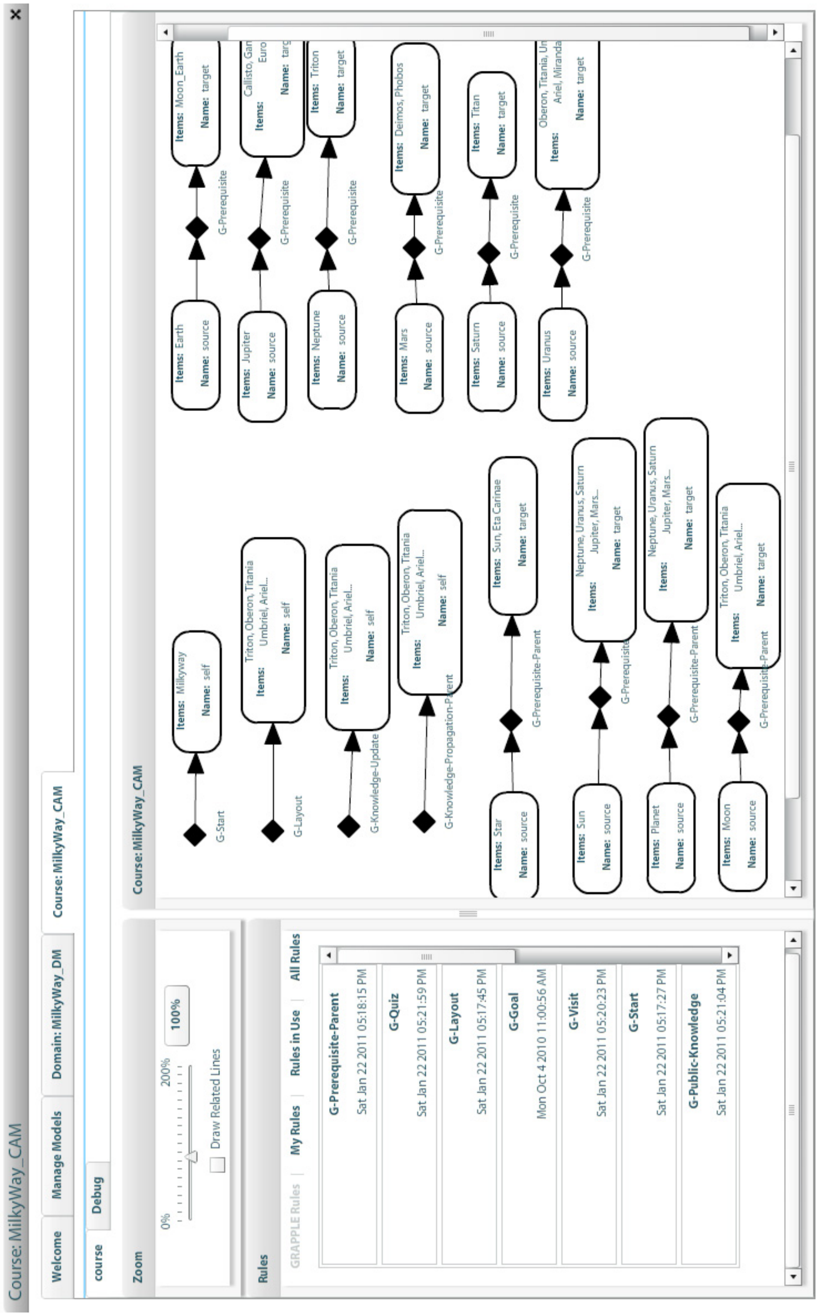


Fig. 6.4 The Course Tool (showing the Milkyway course)

- G-Start defines which concept is the first concept every learner should study first; (although in theory you could define multiple starting points there would be no guidance to help the learner decide which concept to start with),
- G-Layout defines the presentation of the concepts, typically including which *view* is used to show a navigation menu; different concepts may be presented using a different layout, and layout can even be adaptive,
- G-Knowledge-Update states that studying a concept implies that the learner acquires knowledge about that concept,
- G-Knowledge-Propagation-Parent states that knowing something about a concept implies that you also know a bit about its parent concept,
- G-Prerequisite makes the “target” concepts *suitable* when the learner has enough knowledge about all the “source” concepts (and *G-Prerequisite-Parent* does the same when a parent concept is a prerequisite for its children).

As Fig. 6.4 shows, the pedagogical relationships are all “separate”. You cannot create a *sequence* by pasting prerequisites together. Some authors may find this troublesome because they have an adaptive sequencing mind-set rather than an adaptive hypermedia mind-set (for informal learning based on exploration).

What figures 6.3 and 6.4 do not show is that the Domain and Course Tools can be placed side by side and concepts can be copied/pasted or dragged/dropped from the Domain to the Course Tool. Although the tools show “flexibility” there is no adaptive behavior in these tools: authors define adaptation for the learner but experience no adaptation during the authoring process.

Fig. 6.4 does not show how the PRTs actually work to update the UM and to cause the ALE to perform adaptation. We describe this in the next section which is more technical. Authors can use PRTs without understanding the implementation details of PRTs. A non-technical reader of this chapter may opt to skip the next section.

6.3.3 Pedagogical Relationship Types

PRTs connect one or more sets of concepts (which we call *sockets*). Our examples show PRTs with one or two sockets, but, as said, there can be in general arbitrarily many. We will study a few PRTs to explain how UM updates and adaptation are defined to work with GALE.

G-Knowledge-Update and *G-Knowledge-Propagation-Parent* work together to register knowledge as an “effect” of reading pages. (Knowledge can also be registered as a result of multiple-choice tests, which we do not describe in this chapter). A typical approach to registering knowledge is to consider that knowledge is “propagated” through a concept hierarchy that is defined through “parent” (DM) relationships. It’s like knowledge of a chapter being built up from knowledge of the underlying sections, and knowledge of sections being built up from knowledge of the underlying subsections or pages.

To this end the PRTs define two UM variables (called *attributes* in GALE): *knowledge* and *own_knowledge*. The *own_knowledge* is obtained by studying a concept alone, i.e. by reading the corresponding page. We use two knowledge levels: 100 signifies complete knowledge and 35 signifies partial knowledge. (In the actual implementation a %level% parameter is used instead of the fixed number 35.) The *knowledge* of a concept is calculated by considering the *own_knowledge* and the *knowledge* of all underlying concepts combined. To this end the *G-Knowledge-Update PRT* uses the following (simplified) GALE code:

```
%self% {
  event + `
    if (${#suitability}) {
      #{#own_knowledge, 100};
    }
    if (!${#suitability} && ${#own_knowledge} < 35)
      #{#own_knowledge, 35}; `

  #own_knowledge { event + `
    #{#knowledge, ${#knowledge}+changed.diff/(${<-
(parent)}.length+1)};  `}
}
```

The template[12] code consists of two parts. The top part is triggered by the “event” of accessing the concept. The *own_knowledge* of the concept is set to either 100 or 35 depending on the *suitability* of the concept (and the previous value). This rule is executed by GALE’s adaptation engine and the resulting UM update is sent to a UM service. The bottom part is triggered by the event of *changing* the *own_knowledge* (which is done by the top part). The change to the *own_knowledge* contributes to the *knowledge* attribute, but only fractionally (by dividing by the number of children + 1). If a concept has 4 children, for instance, then reading its page contributes only 20% to the *knowledge* attribute as each child also contributes 20%. The rules in the bottom part are executed by the UM service that acts as an *active database*, using ECA rules.

To understand the knowledge updates completely we also need to look at the code for *G-Knowledge-Propagation-Parent*:

```
%self% {
    #knowledge {event + `
        #{->(parent)#knowledge, ${->(parent)#knowledge}+
         changed.diff/(${->(parent)<-
(parent)}.length+1)};  `}
}
```

This code is triggered by a change to *knowledge* (generated by *G-Knowledge Update* for instance) and it propagates a fraction of the knowledge update to the parent concept. This rule is executed by GALE’s UM service.

[12] We call it *template* code because it contains placeholders for concepts that are instantiated with real concepts when a PRT is used in the Course Tool.

G-Prerequisite (and *G-Prerequisite-Parent*) sets the *suitability* of a concept to true or false depending on the *knowledge* of its prerequisites. Actually this PRT can set the *suitability* for a number of concepts at once as both the source and target sockets may contain multiple concepts. The (simplified) code is:

```
%target% {
    #suitability & !`(${%source%#knowledge}>70)`
}
```

For each of the concepts in the *source* socket the *knowledge* value must be over 70 (or a parameter %level% in the complete implementation) in order for the *target* concept(s) to become *suitable*. The & symbol denotes the Boolean "and" operator and the ! symbol denotes that the following code must be executed to result in a value. The difference with the previous examples was that they contained code that was triggered by events, hence no ! symbol to force execution to return a value. The computation of the *suitable* value is done by the UM service in GALE.

We do not have space to describe more PRTs in detail, but we hope that by showing these few examples we give an impression of how detailed UM updates and adaptation (actually setting values like *suitability* that are used by GALE to perform adaptation) can be specified. An advanced adaptation author or designer can define arbitrarily many new PRTs and associate code with them. The code can define and make use of arbitrarily many UM attributes with arbitrarily chosen names. (GALE will create all the used UM attributes.)

From the discussion above we observe that the "intelligence" of the adaptation in GRAPPLE comes from a combination of defining adaptation rules (actually PRTs) at the *concept* level and from connecting concepts through PRTs by means of the Course Tool. When using the pre-defined PRTs an author is (almost) guaranteed to end up with sensible adaptive behavior and also to avoid infinite loops (of rules triggering each other). By defining one's own PRTs and associated GALE code one can define any desired behavior, whether it makes sense for an educational application or not. The GALE code is so general that it can equally well be used to turn navigating a course into an adventure game that drives the end-user mad. In order to obtain meaningful evaluation results about the use of GRAPPLE in different educational settings all evaluation experiments have made use of only the pre-defined PRTs. Creating new PRTs is a too complex task to try this in reasonably small/short experiments.

6.4 The GRAPPLE Infrastructure

The adaptation as described in Sect. 6.2 and 6.3 suggests that UM updates and adaptation only depend on actions the learner performs when interacting with a GALE course. As GRAPPLE aims to support life-long learning it provides an infrastructure through which different LMSs can interact with a shared (distributed) user model framework (GUMF) that is used by GALE as an extra source of user information on which to base its adaptation. We show the architecture in Fig. 6.1, and now provide an explanation of the components and how they work together.

The top left of Fig. 6.1 shows the Shibboleth identity provider. In order to have life-long learning support the learner should have an identity that can be used everywhere: on different LMSs and on GALE (possibly different instances of GALE as well). Moving to a different institute or company can be done without losing the user model stored in the shared GUMF service. (Recently OpenID[13] is gaining popularity and could be considered as an alternative for Shibboleth in the future.)

GRAPPLE uses a common asynchronous bus through which components can send messages to each other (without blocking while waiting for a possible answer). Here is a possible scenario:

Let us assume you are taking a course administered by an LMS, with adaptive course material served by GALE. When you log in on the LMS you will see some kind of link or menu item to take you to the adaptive course text. The course text appears either within the same browser window or in a separate window, depending on how the LMS is set up (so possibly providing a tightly integrated presentation in which you may not even be aware that the course text is not really being served by the LMS). The adaptation in GALE partly depends on your interaction with the course text, partly on information retrieved from GUMF, from courses you took before, and partly on what you do in this course with the LMS. You may for instance take a multiple-choice test on the LMS in order to move from an introductory to an advanced course part. The LMS stores your test score in the "grade book" but also sends it to GUMF. GUMF does not forward this information to GALE (immediately), but when your navigation in the course text reaches the advanced part GALE may (be configured to) request your test score from GUMF in order to decide to grant or deny you access to the advanced part, or possibly to adapt in some other way, like directing you to remedial learning material. Remedial learning material may adaptively be offered to learners who failed a specific test in a different course that was taken earlier (and probably even passed). Two important things to note here:

- A conversion part of the authoring interface (not shown in Sect. 6.3) is needed to convert the scale (for test results) used by the LMS to the scale (for knowledge values) used by GALE. A different conversion may be needed for scores coming from different LMSs. (Fig. 6.1 shows the GRAPPLE Conversion Component, GCC, in the LMS block.),
- When you go through a remedial learning phase to increase your knowledge about a certain topic your improved knowledge state is communicated by GALE to GUMF (and later used in adaptation) but the test score on the LMS does not change (and neither does a grade stored in a grade book on the LMS). So in GRAPPLE we can distinguish between grades that are used "officially" (that are in the grade book and may appear on report cards) and knowledge that is obtained (possibly later than the test) and used for adaptation.

[13] openid.net.

Preparing an LMS for GRAPPLE mainly consists of three simple steps:

- The LMS needs to be told the address of the event bus GEB. Fig. 6.5 below shows this dialog for the Claroline LMS. This dialog also includes a second address: that of the web service to retrieve addresses of adaptive course texts available through this GRAPPLE installation. When creating links to adaptive course texts another user interface is used that lets you select a course from that list (see Fig. 6.7 below),
- Each LMS "tracks" certain events that potentially generate interesting information about the learner. Depending on which information is used for adaptation you may or may not wish to have such event information forwarded to the GUMF user model service. Fig. 6.6 below shows the dialog for selecting information to be sent to GUMF as designed and implemented for the Sakai LMS,
- In every LMS you can add "resources" to a course. GRAPPLE adds the "GRAPPLE" resource type. Because the address of an adaptive course is long and complex a dialog was added to select courses from a list retrieved through the above mentioned web service. Fig. 6.7 shows this dialog for Moodle.

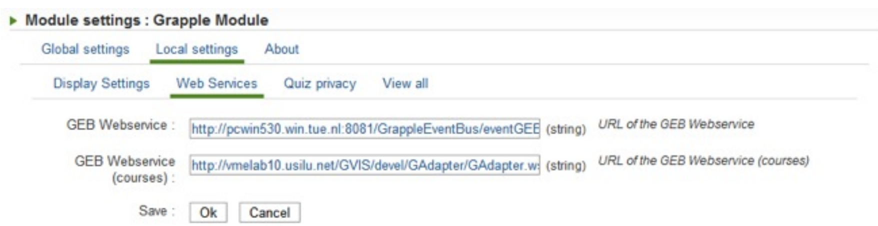


Fig. 6.5 Connecting Claroline to GRAPPLE (to GEB)

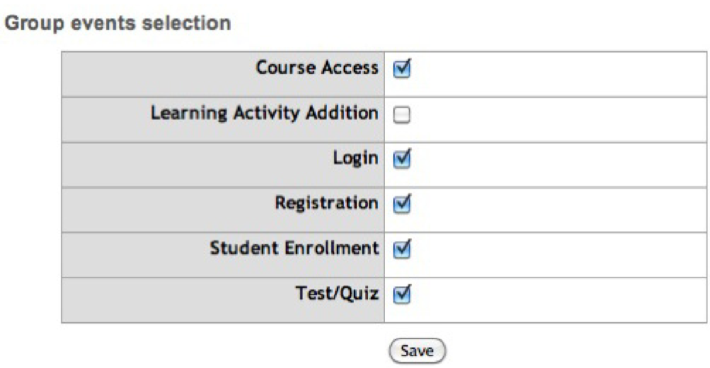


Fig. 6.6 Event selection for Sakai to GUMF communication

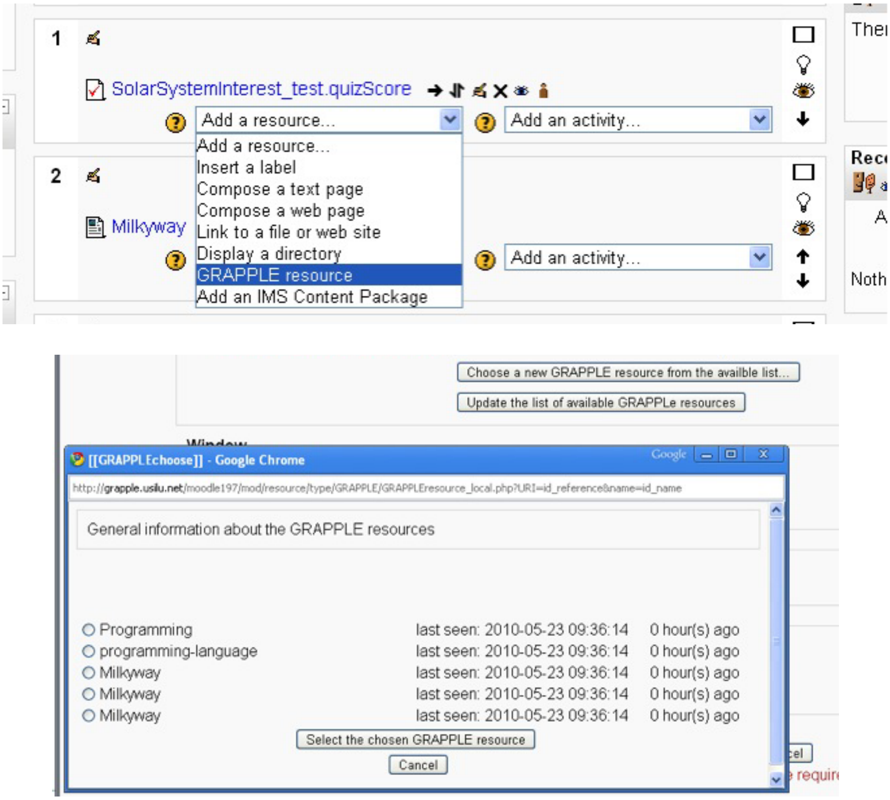


Fig. 6.7 Dialog for selecting an adaptive course in Moodle

6.5 Teaching and Learning Experience Using Adaptation

The GRAPPLE infrastructure has been used in numerous experiments and courses by several GRAPPLE project partners over the past two years (i.e. since the first parts of the technology became available). We only report on a few experiments and findings here.

Learning through *adaptive information exploration* is first and foremost intended for individual self-paced learning, but it can also be used to provide the equivalent of a textbook that is used as study material to accompany a regular course with lectures and labs. Unlike a lecture the adaptive course text enables the learner to follow their own path, adapted to their learning style (e.g. by offering examples before theory or vice versa), and adapted to their foreknowledge (by offering prerequisite learning material to learners who need it and leaving that out for students who have sufficient prior knowledge). The core of the learning material remains the same but the learner has more *navigational freedom* than in lectures, a traditional textbook or *guided tours*. The aim of an adaptive course is to have all learners reach the same knowledge and skill level at the end of a course. This is not only a requirement for university courses but also for in-company training programs (often resulting in a certificate). Instead of achieving this by forcing all learners to study every detail the course text has to offer and by making them take all the tests we rely on (and trust) information about prior knowledge as indicated by GUMF. It should be clear that instructors, institutes and companies sharing a GRAPPLE environment should agree on how much trust they place on each others' information that is stored in GUMF.

6.5.1 The Milkyway Example

A master student at the TU/e developed an example adaptive course about celestial objects (Ploum 2009). This course has been used for different purposes:

We presented the Milkyway DM to students and workshop participants (on numerous occasions) who come up with a *start concept* and with *prerequisites* to be used when developing a Course Model. To make the DM clearer than what the DM tool shows we used a paper (A3 size) presentation shown in Fig. 6.8. We referred to these workshops in Sect. 6.2 as they revealed the need for PRTs with *sockets* that can hold multiple concepts, to make drawing of the pedagogical structure easier.

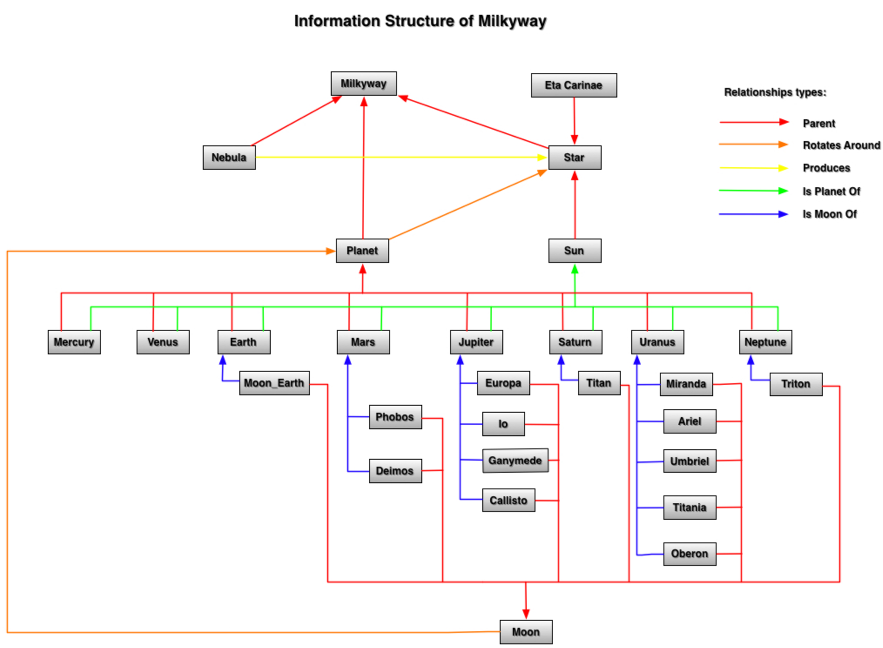


Fig. 6.8 The Milkyway DM, used in "prerequisite" workshops

Experience from these workshops has taught us that authors tend to *overspecify* prerequisites (for instance ignoring that prerequisites are transitive: if A is a prerequisite for B and B for C then A is automatically a prerequisite for C so that need not be specified in the model), and that authors often start out by thinking in terms of a *sequence* or *learning path* instead of leaving as much navigational freedom as possible in their course and only using prerequisites where really needed. (In later workshops we started with a more explicit introduction about the difference between adaptive sequencing and adaptive hypermedia.).

We used the Milkyway example also to introduce the notion of using *template pages* instead of creating each course page by hand. Fig. 6.9 shows the difference between the two authoring processes:

- On the one hand writing (or importing) individual pages separately is easy, as it only requires basic knowledge of XHTML (or a web page authoring tool). On the other hand it requires discipline to ensure that all pages that should have the same layout indeed do, and it involves repetitive work if later that layout needs to be changed (everywhere). In authoring experiments some participants have *generated* pages from Wiki sites. The consistent layout already comes from that generation process so the "writing" individual pages approach works well in this case,

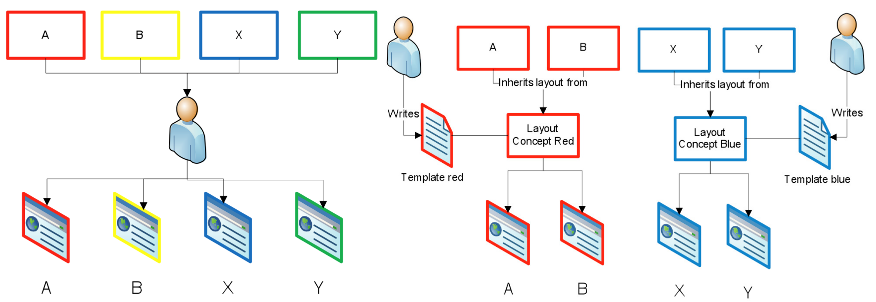


Fig. 6.9 Writing individual pages (left) versus using template pages (right)

- Using template pages ensures that through a single template all pages that should have the same layout and presentation are consistent, and any change to the layout automatically applies to all pages based on the same template.

A typical page of the Milkyway course is shown in Fig. 6.10. This page describes the planet Jupiter. Pages describing other planets look very similar because they are all based on the same template.

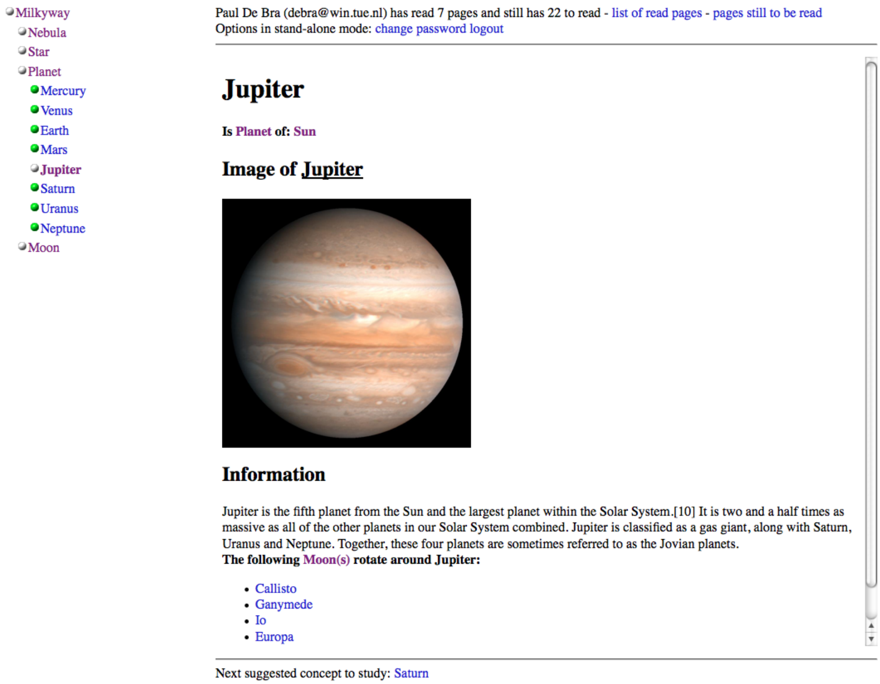


Fig. 6.10 A typical page from the Milkyway course

Some parts are included because of a global *layout* definition. These include the navigation (accordion) menu on the left, the header, the footer and the “page” (or the *resource* of the requested concept, to be exact). In the navigation menu the status (suitability) of links is not only indicated through the link color but also through colored balls that were first introduced in ELM-ART (Brusilovsky et al. 1996) and later used in other systems such as Interbook (Brusilovsky et al. 1998) and KBS-Hyperbook (Fröhlich et al. 1998). The structure of the “page” is based on a template, which is an XHTML file with some GALE tags added to it, which leads us to the third type of usage of Milkyway: GALE tags and code.

The Milkyway example makes extensive use of GALE tags added to the XHTML page. In these tags expressions are used that refer to the Milkyway DM to present information that pertains to the “current” concept:

- The title “Jupiter” is generated by including the *title* property of the current concept. (The <variable> tag is used here.),
- The “Is Planet of: Sun” statement is generated by means of the *title* of the *parent* concept (Planet) and the *title* of the concept to which the current concept has an *isPlanetOf* relationship. (Expressions are used with the <variable> and the <a> tags.),
- The *title* is used again to produce the text “Image of Jupiter”,

- The image is chosen by means of the *image* property of the current concept. (The <attr-variable> tag is used to insert the image URL into the <img> tag.),
- The information paragraph is an included <object> that is chosen based on the *info* property of the current concept,
- "The following Moon(s) rotate around Jupiter" makes use of existence of concepts with an *isMoonOf* relation to the current concept (and of the *title* of the current concept to show "Jupiter" again). (The <if> tag is used to include this sentence or not depending on the existence of moons.),
- The list of moons is generated by means of the <for> tag, looping over the concepts that have an *isMoonOf* relation to the current concept, and using the *title* property of each of them. The fact that it's a bullet list is simply coded as XHTML in the template page. The list could be presented in any other way as well.

Apart from the Milkyway example we also created a GRAPPLE tutorial that not only explains almost everything about GRAPPLE but that is an adaptive course itself, thus illustrating the use of the GRAPPLE infrastructure. The GRAPPLE tutorial has been created by writing each page separately. Since many authors taking part in the GRAPPLE experiments were shown both Milkyway and the GRAPPLE tutorial they got to experience a template-based and a non-templated-based example that can be used to learn about authoring "by example".

6.5.2 Authoring Experiments

Most GRAPPLE partners (both academic and corporate) held authoring workshops and/or gave authoring (group) assignments. The experiments were run with experts in learning content creation in industry, students and teachers (in higher education). For all experiments the subjects first received some basic training about adaptive technology-enhanced learning and about GRAPPLE, most specifically about GAT and GALE. They were then asked to develop a course, either on a given topic or a topic of their own choice. Example courses were developed on huge variety of topics: self-management, tying knots, sex and evolution, hot and cold drinks, Star Trek Voyager, tennis, PHP, web technology, the animal kingdom, Knowledge Space Theory, hypermedia, adaptive hypermedia, etc., etc.. The courses varied from around 20 concepts (the knots) to well over 100 concepts (sex and evolution). Some interesting findings from these experiments are:

1. The cost of adding adaptation to an already existing on-line (hypermedia) course varies greatly. Students at the TU/e estimated this to represent around 15 hours of work for the "courses" they created. The time needed to convert an existing course without on-line course material into an on-line adaptive course was also estimated in authoring pilots at the Netherlands Open University and the University of Graz and there it was found to be very high: at least 6 months full time for a 1 ECTS module (representing 25 to 30 hours of study time). These estimates are so wildly different that no definite conclusions can be drawn from them. The TU/e students were finishing a course on adaptive systems so they were well trained in the task at hand whereas the participants in

the other workshops were novices in the area of adaptive hypermedia. Once authors become more experienced we expect them to need much less time than the most pessimistic numbers we found,

2. Assigning multiple resources (learning objects) to multiple concepts led to an undesirable result; learning objects were included multiple times in the course. It is therefore recommended to achieve a 1:1 relation between concepts and learning objects. An alternative solution would be to create a PRT that can deal with this situation (to ensure each learning object is only presented with the concept that is visited first by the learner). No such PRT was available in the predefined set, so the 1:1 modeling option was used,
3. For a large DM the DM tool requires too much repetitive work. Also, the visual presentation of a DM by the tool is not very attractive. (Compare Fig. 6.2 from the DM tool with Fig. 6.5 for instance, created with a plain office program.) Some people therefore used different drawing or mind-mapping tools and converted their output to IMS VDEX[14], the format used by the DM tool. This experience stresses the need for more converters to be part of the GRAPPLE distribution,
4. The GAT authoring environment did not score well on the aspect of usability of the interface. Our attempt to clearly separate the DM (relating to the subject domain) from the Course Model (relating to the pedagogical domain) was not appreciated. Users would have been happier with a single integrated tool[15]. This suggests that when creating a course authors do not mentally distinguish between topic relations and pedagogical relations,
5. The Course Tool posed some performance problems for large models. This problem was solved, but too late for most experiments so it has influenced the lower usability scores,
6. Some example courses used content taken from existing wiki sites. Importing content from (the output of) other applications leads to GALE applications with non-template pages. Some projects done at the TU/e used template pages. Not surprisingly all these projects were done by computer science students,
7. GALE was found to be easy to use. However, with the default simple layout and style-sheet templates the presentation was not found to be attractive. Fortunately this can be fully configured and even made adaptive, so it is not a limitation in the architecture, only in the authoring effort involved.

6.5.3 Learning Experiments

Some of the developed adaptive courses were used in learning experiments, to check whether adding adaptivity to on-line learning is beneficial to the learner. Previous research on evaluating learning through adaptive systems (Brusilovsky et al. 2004, Hsiaom et al. 2010, Weibelzahl 2001) is plentiful but does not deliver an

[14] www.imsglobal.org/vdex/

[15] In the older AHA! system creating the DM and Course Model was combined in a single graphical tool and the experience with that prompted the separation that was made in GAT, so clearly there are advantages and drawbacks to both approaches.

answer to the question “does adaptation help” because that answer is very dependent on how the adaptation is actually used. The question is meaningless in the context of GRAPPLE in general, but still some experiments were conducted, using the standard pre-defined PRTs (mostly prerequisites).

From several learning experiments we learned that the adaptation provided by the system (most visible was the link annotation using link colors and colored balls as shown in Fig. 6.7) was considered helpful for learning (score 85%). However, a larger scale experiment where the same subject was taught to two groups of students, one in the traditional way and one through an adaptive on-line version, was not found to yield better learning outcome. The ability to learn when and where the learner wanted was appreciated, but that is a consequence of offering learning material on-line, not of making it adaptive.

Below we provide some detailed information from a study conducted at the TU/e to investigate several aspects of learning by means of a course offered through GALE. The course used was the GRAPPLE tutorial. Students were asked to complete a questionnaire with the following questions:

- Ease of use of GALE: Half of the students found GALE easy to use, partly thanks to the recommendations it offers (link annotations, navigation menu and guided tour). Others found it not so easy or had difficulty understanding the content (which is not really a GALE issue but due to how the course text explains the subject). One student complained about the lack of *scrutability* of the adaptation strategy. This is a valid comment: it is possible to add scrutability of the user model to a course but not of the adaptation,
- Usefulness of GALE: In which aspects may the system enhance or handicap learning performance? Over 80% found GALE to be useful, including the use of link annotations, content adaptation and tests to validate the learner’s knowledge. Others were either indecisive or would have preferred different adaptation rules (which were not further specified),
- Intention to use: Given a choice, would you use GALE in the future? Over half of the students would definitely or possibly use GALE in the future. To get a better idea why students did or did not like working with GALE we gathered some more comments. The *prerequisites* and resulting advice were appreciated. Negative comments dealt with aspects of the specific course and presentation and not with inherent properties of GALE:

  – Layout is not nice/layout should be improved: Layout can be configured in every possible aspect in GALE. The authors of the GRAPPLE tutorial used in the experiment opted for a simple layout similar to that of the Milkyway example shown in Sect. 6.5. It is definitely possible to create a more attractive layout. In GALE it is even possible to make the layout adaptable or adaptive (automatically adapted to properties in the UM),
  – Critique on course content: Clearly adaptation is not a magic wand that turns bad content into highly readable course material that is easy to understand. Some students had trouble understanding the GRAPPLE tutorial used in the experiment,

  - Insufficient function: Although students were not asked to provide more detail about this comment we can acknowledge that the course makes use of only the predefined PRTs like *prerequisites*, *knowledge update* and *knowledge propagation (parent)* and thus only uses a small fraction of the adaptation possibilities GALE offers. But apart from that students may also have missed a *search* function, *scrutability* and *concrete guidance* like the “teach me” option in Interbook (Brusilovsky et al. 1998).

- Pros and cons of using GALE in university courses: When asked specifically about the university context students were positive (just over half) about the adaptation, the flexibility (navigation freedom) and the use of adaptive course material to complement traditional university courses.
- Negative remarks were that the adaptation to the user might be difficult because of differences between users, that users get the “feeling” of being watched, that internet access is required in order to access the course text, that students may have a preference for printed material, and that it is a problem that questions from students cannot be answered directly,
- Suitability of GALE for which application scenarios in a university:: Students would recommend using GALE mostly for simple or small courses, for learning and testing course material, for distance education (courses that do not require face-to-face attendance), for refreshing or reviewing learning content, and for courses with a lot of literature,
- Opinions on the adaptation provided in the course: One third of the students commented that they found the guidance useful, whereas almost as many students commented that they did not notice the adaptation. Of course they must have noticed changes in link color (and the colored balls) but as links changing from blue to purple is normal browser history behavior these students did not associate that behavior with adaptation. Also, content adaptation in the GRAPPLE tutorial is very subtle and can easily go unnoticed. Some individual comments were on the lack of scrutability; some students were irritated that their (significant) foreknowledge did not result in adaptation (e.g. skipping known topics) and some students did not like the adaptation rule that considers a concept to be studied when it has simply been accessed. Indeed, the specific course (GRAPPLE tutorial) did not use or test for foreknowledge in order to allow knowledgeable students to skip some introductory topics, although the GRAPPLE infrastructure would certainly allow for this, and the course used the “access implies knowledge” principle instead of “passing a test implies knowledge” principle. We have made more extensive use of tests for adaptation in another long running course (GALE is certainly up to this task.),
- What is the potential of GALE to support lifelong learning? Several students remarked that “further improvements should be made” or saw limited potential as GALE cannot build a comprehensive UM on all the knowledge of a user. We acknowledge that through the small scale experiments the students could not experience any of the functionality intended to support life-long learning, so any potential they would see came from studying GRAPPLE (through the tutorial), not from experiencing it.

Overall the learning experiments were a positive evaluation of the GRAPPLE technology (more so than authoring experiments) but they did not give us much insight beyond what we already knew: during a learning experiment students consider that the functionality they experience is all the functionality the system has to offer (so they fail to grasp the potential they do not see); they may not see adaptation because they only have their own experience (and do not see that another student with a different history is presented with different learning material); they also do not grasp what life-long exposure to this technology and to a common user model would enable. They do experience learning through an adaptive on-line course and generally find this to be a flexible form of distance learning.

6.6 Conclusions and Future Work

In this chapter we have introduced the GRAPPLE infrastructure that is aimed at providing life-long learning through a combination of LMS and ALE. We have presented GAT, a tool for authoring the conceptual and pedagogical structures of a course and GALE, the adaptation engine that presents adaptive learning material to the learner. GRAPPLE also offers a shared user model framework GUMF that has not been used extensively in the authoring and learning experiments so far. This part of the infrastructure will play an important role in the next series of experiments that will progress towards the “life-long” aspect of the learning. Instead of registering course grades in a “gradebook” we intend to store a finer-grained user model in GUMF, containing at least test results on individual course topics. Follow-up courses (in the same university or company or in a different one) can tap into GUMF to decide which smaller topics are not mastered to a sufficient level and to then adaptively offer remedial learning material. In this way we intend to improve the learners’ knowledge level after a course has ended and avoid that learners fail in follow-up courses because of missing foreknowledge. We are hopeful that adaptively offering remedial learning material will result in improvements of the learning outcome, something that adaptation restricted to the user information gathered during a single course has not yet demonstrated.

The real bottleneck in large scale adoption of adaptive learning is still the authoring of the adaptation, or “intelligence” of the ALE. GRAPPLE attempts to solve this problem by means of PRTs (Pedagogical Relationship Types). Each PRT has a bit of intelligence (adaptation rules), and an author just combines instances of PRTs into a complete course model. Using the predefined PRTs (like *knowledge update* and *prerequisite*) this is a straightforward authoring task that more or less guarantees a sensible adaptive outcome, as is evidenced by the large number of example courses that were developed successfully during a series of authoring experiments. However, once an author starts designing his/her own PRTs the intelligence can easily “go wrong”. GALE cannot decide whether the implemented intelligence is actually what an author intended but has a built-in time-out to abort infinite loops in adaptation rule sequences. In the future we need to add a static analysis method to detect adaptation errors. In (Wu and De Bra 2001, Wu et al. 2001) it is already shown that such analysis cannot detect all problems with adaptation rules, but we hope that this analysis can capture the most common errors at authoring time.

Acknowledgments. This chapter describes work performed in the EU IST FP7 project GRAPPLE, project 215434. It could not have been written without the support of the entire GRAPPLE team putting the entire infrastructure together.

References

Abel, F., Henze, N., Herder, E., Krause, D.: Interweaving Public User Profiles on the Web. In: De Bra, P., Kobsa, A., Chin, D. (eds.) UMAP 2010. LNCS, vol. 6075, pp. 16–27. Springer, Heidelberg (2010)

Albert, D., Nussbaumer, A., Steiner, C.M., Hendrix, M., Cristea, A.I.: Design and development of an authoring tool for pedagogical relationship types between concepts. In: Proceedings of ICCE (2009)

Brusilovsky, P.: Methods and techniques of adaptive hypermedia. User Modeling and User Adapted Interaction 6, 87–129 (1996)

Brusilovsky, P.: Adaptive hypermedia. User Modeling and User Adapted Interaction 11, 87–110 (2001)

Brusilovsky, P., Eklund, J., Schwarz, E.: Web-based education for all: A tool for developing adaptive courseware. In: Proceedings of IWWWC, vol. 30(1-7), pp. 291–300 (1998)

Brusilovsky, P., Karagiannidis, C., Sampson, D.: Layered evaluation of adaptive learning systems. International Journal on Continuous Engineering Education and Lifelong Learning 14(4/5), 402–421 (2004)

Brusilovsky, P., Schwarz, E., Weber, G.: ELM-ART: An intelligent tutoring system on world wide web. In: Proceedings of ITS, pp. 261–269. Springer, Heidelberg (1996)

Cristea, A.I., de Mooij, A.: The three Layers of adaptation granularity. In: Proceedings of UM, p. 145. Springer, Heidelberg (2003)

de Bra, P., Smits, D., Sluijs, K., van der Cristea, A.I., Hendrix, M.: GRAPPLE: Personalization and adaptation in learning management systems. In: Proceedings of ED-MEDIA (2010)

Fröhlich, P., Nejdl, W., Wolpers, M.: KBS-Hyperbook – An open hyperbook system for education. In: Proceedings of ED-MEDIA (1998)

Gutierrez-Santos, S., Pardo, A., Delgado Kloos, C.: Authoring courses with rich adaptive sequencing for IMS learning design. Journal of Universal Computer Science 14(17), 2819–2839 (2008)

Hendrix, M., Cristea, A.I., Stewart, C.D.: Adaptation languages for learning: the CAM meta-model. In: Proceedings of the IEEE-ICALT, pp. 104–106 (2009)

Hendrix, M., De Bra, P., Pechenizkiy, M., Smits, D., Cristea, A.I.: Defining Adaptation in a Generic Multi Layer Model: CAM: The GRAPPLE Conceptual Adaptation Model. In: Dillenbourg, P., Specht, M. (eds.) EC-TEL 2008. LNCS, vol. 5192, pp. 132–143. Springer, Heidelberg (2008)

Hendrix, M., Cristea, A.I.: Design of the CAM model and authoring tool. A3H. In: Proceedings of 7th Authoring of Adaptive and Adaptable Hypermedia Workshop of ECTEL (2009)

Hsiaom, I., Brusilovsky, P., Yudelson, M., Ortigosa, A.: The value of adaptive link annotation in E-Learning: A study of a portal-based approach. In: Proceedings of ACM-CHH, pp. 223–228 (2010)

Knutov, E., De Bra, P., Pechenizkiy, M.: AH 12 years later: A comprehensive survey of adaptive hypermedia methods and techniques. New Review of Hypermedia and Multimedia 15(1), 5–38 (2009)

Ploum, E.: Authoring of adaptation in the GRAPPLE project. Master Thesis, Eindhoven University of Technology (2009)

Smits, D., De Bra, P.: GALE: A highly extensible adaptive hypermedia engine. In: Proceedings of ACM-CHH, pp. 63–72 (2011)

Stash, N., De Bra, P., Cristea, A.I.: Adaptation to learning styles in a general-purpose system AHA (Adaptive Hypermedia Architecture). Educational Technology & Society 11(1) (2008)

Stash, N., Cristea, A.I., De Bra, P.: Learning Styles Adaptation Language for Adaptive Hypermedia. In: Wade, V.P., Ashman, H., Smyth, B. (eds.) AH 2006. LNCS, vol. 4018, pp. 323–327. Springer, Heidelberg (2006)

Van der Sluijs, K., Höver, K.: Integrating adaptive functionality in a LMS. International Journal of Emerging Technologies in Learning 4(4), 46–50 (2009)

Weibelzahl, S.: Evaluation of Adaptive Systems. In: Bauer, M., Gmytrasiewicz, P.J., Vassileva, J. (eds.) UM 2001. LNCS (LNAI), vol. 2109, pp. 292–294. Springer, Heidelberg (2001)

Wu, H., De Bra, P.: Sufficient Conditions for Well-Behaved Adaptive Hypermedia Systems. In: Zhong, N., Yao, Y., Ohsuga, S., Liu, J. (eds.) WI 2001. LNCS (LNAI), vol. 2198, pp. 148–152. Springer, Heidelberg (2001)

Wu, H., de Kort, E., De Bra, P.: Design issues for general-purpose adaptive hypermedia systems. In: Proceedings of ACM-CHH, pp. 141–150 (2001)

Abbreviations

ALE	Adaptive Learning Environment
CAM	Conceptual Adaptation Model
DM	Domain Model
ECA	Event-Condition-Action
GALE	GRAPPLE Adaptive Learning Environment
GAT	GRAPPLE Authoring Tool
GCC	GRAPPLE Conversion Component
GRAPPLE	Generic Responsive Adaptive Personalized Learning Environment
GUMF	GRAPPLE User Modeling Framework
HTTP	HyperText Transfer Protocol
LMS	Learning Management System
PRT	Pedagogical Relationship Type
TU/e	Eindhoven University of Technology
UM	User Model
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
XHTML	eXtensible HyperText Markup Language
XML	eXtensible Markup Language

Chapter 7
Performance Evaluation of Decision-Based Content Selection Approaches in Adaptive Educational Hypermedia Systems

Pythagoras Karampiperis[1,2] and Demetrios G. Sampson[2,3]

[1] National Center of Scientific Research "Demokritos"
P. Grigoriou & Neapoleos Str., GR-15310 Aghia Paraskevi Attikis, Greece
pythk@iit.demokritos.gr
[2] Department of Digital Systems, University of Piraeus
150 Androutsou Street, Piraeus, GR-18534, Greece
[3] Centre for Research and Technology Hellas
6th Klm. Charilaou - Thermi Road, Thermi, Thessaloniki, GR - 57001, Greece
sampson@iti.gr

Abstract. Adaptive content selection is recognized as a challenging research issue in adaptive educational hypermedia systems (AEHS). In order to adaptively select learning objects (LO) in AEHS, the definition of adaptation behavior, referred to as Adaptation Model (AM), is required. Several efforts have been reported in literature aiming to support the AM design by providing AEHS designers with either guidance for the direct definition of adaptation rules, or semi-automated mechanisms which generate the AM via the implicit definition of such rules. The goal of the semi-automated, decision-based approaches is to generate a continuous decision function that estimates the desired AEHS response, aiming to overcome the problems of insufficiency and/or inconsistency in the defined adaptation rule sets. Although such approaches bare the potential to provide efficient AM, they still miss a commonly accepted framework for evaluating their performance. In this chapter, we discuss a set of performance evaluation metrics that have been proposed by the literature for validating the use of decision-based approaches in adaptive LO selection in AEHS and assess the use of these metrics in the case of our proposed statistical method for estimating the desired AEHS response.

7.1 Introduction

AEHS have been proposed as the underlying facilitator for personalized web-based learning with the general aim of personalizing learning experiences for a given learner (De Bra 2006, Knutov et al. 2009).

A. Peña-Ayala (Ed.): Intelligent and Adaptive ELS, SIST 17, pp. 161–182.
springerlink.com

In order to adaptively select and sequence LO in AEHS, that is, content objects described with educational metadata (McGreal 2004, Harman and Koohang 2006), the definition of adaptation behavior is required (Nejdl and Brusilovsky 2004). The AM contains the rules for describing the runtime behavior of the AEHS. In the literature, there exist different approaches aiming to support the AM design by providing AEHS designers with either guidance for the direct definition of adaptation rules, such as Authoring Task Ontology - ATO (Aroyo and Mizoguchi 2004), My Online Teacher - MOT (Cristea and Kinshuk 2003, Cristea 2007) and ACCT (Dagger et al. 2005), or semi-automated mechanisms which generate the AM via the implicit definition of such rules (Karampiperis and Sampson 2005, Huang et al. 2008, Ras and Ilin 2008).

The main drawback of the direct definition of adaptation rules is that there can be cases during the run-time execution of AEHS where no adaptation decision can be made due to insufficiency and/or inconsistency of the defined adaptation rule sets (Wu and De Bra 2001, Brusilovsky et al. 2007). This is due to the fact that, even if appropriate resources exist in the media space, the absence of a required rule (insufficiency problem) or the conflict between two or more rules (inconsistency problem), prevents the AEHS to select and use them in the generated learning resource sequence. As a result, either less appropriate resources are used from the media space, or required concepts are not covered at all by the resulting sequence (Wu and De Bra 2001).

The goal of the semi-automated approaches is to generate a continuous decision function that estimates the desired AEHS response, overcoming the above mentioned problem (Karampiperis and Sampson 2004). To achieve this, they use data from the implicit definition of sample adaptation rules and attempt to fit the response function on these data. Although such approaches bare the potential to provide efficient AMs, they still miss a commonly accepted framework for evaluating their performance.

This chapter is structured as follows: First, we discuss issues related with the AM design in AEHS focusing on the different approaches used in the literature for the definition of content selection rules. Then, we discuss the performance evaluation metrics that have been proposed by the literature for validating the use of decision-based approaches. Moreover, we present a performance evaluation methodology for decision-based content selection approaches in AEHS, and set up and report simulation-based experiments, following the above mentioned methodology, which aim to validate these evaluation metrics within the framework of our previously proposed statistical method for estimating the desired AEHS response. Finally, we discuss our findings and the conclusions that can be offered.

7.2 Overview of AEHS

Current state-of-the-art AEHS such as AHA! (Stash et al. 2007), OntoAIMS (Aroyo et al. 2003), The Personal Reader (Dolog et al. 2004), WINDS (Kravcik and Specht 2004), ACCT (Dagger et al. 2005) follow an architectural approach that fully implements the core structural elements defined by (Henze and Nejdl 2004) their AEHS definition.

This architecture is a variation of the Adaptive Hypermedia Application Model (AHAM) (De Bra et al. 1999) and consists of two main layers, namely, the run-time layer which contains the adaptation engine that performs the actual adaptation and the design layer. AM design (Brusilovsky and Henze 2007) involves defining:

- Concept selection rules which are used for selecting appropriate concepts from the domain model to be covered,
- Content selection rules which are used for selecting appropriate resources from the media space,
- Sequencing rules which are used for generating appropriate “learning paths” (that is, sequences of LO) for a given learner.

Typically, adaptive educational hypermedia sequencing is based on two main processes, namely, the *concept selection process* and the *content selection process*. In the concept selection process, a set of learning goals from the learning goals hierarchy is selected by the learner e.g. the AIMS (Aroyo and Mizoguchi 2004), or in some cases by the designer of the AEHS e.g. INSPIRE (Papanikolaou et al. 2003). For each learning goal, related concepts from the domain concept ontology are selected. In the content selection process, learning resources for each concept are selected from the media space based on the content selection rules. Typical AEHS examples that utilize this process are the MOT (Cristea and Kinshuk 2003, Cristea 2007), the ApeLS (Conlan et al. 2002), and the ELM-ART (Brusilovsky 2007).

The most commonly used approach for the definition of content selection rules by the AEHS designers team is the direct definition. In this approach, the content selection rules are set by the instructional designer during the design process and they are based on the items of the user model and the resource description model.

As already discussed, the main drawback of the direct definition of adaptation rules is that there can be cases during the run-time execution of AEHS where no adaptation decision can be made due to insufficiency and/or inconsistency of the defined adaptation rule sets. To this end, in the literature, two main approaches have been proposed to overcome these problems.

The first approach uses adaptation patterns (or templates) that have been a priori defined by an instructional designer during the design phase of the AEHS. These patterns hold the content selection rules of the AM. Typical examples of these systems are MOT (Cristea and Kinshuk 2003, Cristea 2007) and ACCT (Dagger et al. 2005).

Although this approach provides a solution to the inconsistency problem, it does not tackle with the problem of insufficiency, since that would require a huge set of patterns, which is difficult to be a priori defined. The problem of defining adaptation rules is a combinatorial problem, which means that in order to design sufficient and consistent adaptation rule sets, all the combinations of the adaptation decision variables should be covered. However, these combinations can be millions (Karampiperis and Sampson 2005), leading to huge rule sets that is difficult to author, manage and verify their sufficiency and/or consistency.

An alternative approach is the use of semi-automated decision based mechanisms (Karampiperis and Sampson 2005, Alfonseca et al. 2007, Huang et al. 2007, Hsieh et al. 2008), which generate a continuous decision function that estimates the desired AEHS response. To achieve this, they use data from the implicit definition of sample adaptation rules and attempt to fit the response function on these data. This definition of implicit adaptation rules, is given in the form of model adaptation decisions, over which the adaptation response function should be fit. This approach overcomes both the problems of sufficiency and consistency; however it introduces decision errors that result from the decision function fitting errors during the machine learning process (Karampiperis and Sampson 2005).

Sect. 7.3 presents the evaluation metrics given in the literature for evaluating the performance of decision-based adaptive content selection and discusses them.

7.3 Performance Evaluation Metrics for Decision-Based AEHS

We focus on the performance evaluation metrics used in semi-automated decision-based approaches for adaptive content selection. Performance evaluation in this context means: measuring how well a semi-automated approach fits the decision function to the provided model adaptation decisions (training data), and how well this decision function responds to decision cases not known during the training process (generalization capacity). As a result, model adaptation decisions are divided into two sets: the training dataset, which is used for evaluating the performance during the training of the semi-automated approach, and the generalization dataset, which is used for measuring the generalization capacity of the approach. Performance evaluation is the comparison result between the expected system output and the estimated AEHS response over the above mentioned datasets.

In adaptive content selection several approaches are proposed in the literature. The most commonly used are the following (Sampson and Karampiperis 2011):

Concept/keyword-based selection: In this approach, searching is made based on keywords representing the desired concepts to be covered from the retrieved LO. In AEHS, these keywords are set over the domain concept ontology at the concept selection process. The ranking of LO is done using a concept/keyword-based similarity formulae (Lee et al. 2006, Biletskiy et al. 2009), which evaluates the relevance of each LO, by comparing the desired concepts/keywords with the classification metadata used for describing the LO in hand.

The main assumption of this approach is that the domain concept ontology and the classification metadata used for the LO share the same concept/keyword terms. However, this is not always true, especially in domains where there exist a variety of classification models which use different terminology for describing a concept depending on the context of use, i.e. in the medical domain there exist many classification systems such as Medical Subject Headings (MeSH), the International Classification of Primary Care (ICPC) etc. targeting different end-users. An alternative approach proposed by (Kiu and Lee 2007), uses unsupervised data-mining techniques for estimating the match between the desired concepts/keywords with the classification metadata used for describing the LO in hand. This approach provides better results from the use of keyword-based similarity formula when different classifications models are used, but it requires significantly more time for the content selection process.

Preference-based selection: In these approaches, selection is performed based on the comparison of the learner profile in hand with the metadata description of the LO. In this case, the ranking of LO is performed using a preference score (Karampiperis and Sampson 2004, Wang et al. 2007, Dolog et al. 2008), which evaluates the utility/suitability of each LO for the learner profile in hand.

In both techniques, the concept/keyword-based and the preference-based selection, general purpose evaluation metrics are used from the field of information extraction (Ochoa and Duval 2008). More specifically, *precision* and *recall* measures are applied in order to evaluate the effectiveness of the LO selection technique, in terms of accuracy and completeness respectively. Precision is the ratio of correct responses to the sum of correct and incorrect responses, and is defined by the equation (7.1) (Wang et al. 2007, Biletskiy et al. 2009):

$$\text{Precision} = \left(\frac{\#\, retrieved\ relevant\ LOs}{\#\, retrieved\ LOs} \right) \tag{7.1}$$

Recall is the number of correct system responses divided by the sum of correct, incorrect and missing system responses, and is defined by the equation (7.2) (Wang et al. 2007, Biletskiy et al. 2009):

$$\text{Recall} = \left(\frac{\#\, retrieved\ relevant\ LOs}{\#\, relevant\ LOs} \right) \tag{7.2}$$

In order to have a single evaluation metric, *F-measure* is used, which is a weighted combination of recall and precision, and is defined by the equation (7.3) (Biletskiy et al. 2009):

$$\text{F - measure} = \left(\frac{2 * precision * recall}{precision + recall} \right) \tag{7.3}$$

However, AEHS implement a content selection strategy which limits the number of retrieved LO, aiming to restrict the amount of information provided to learners at a given time instance, due to the problem of learners' cognitive overload (Brusilovsky 2007). As a result, the precision should be measured not on the entire media space, but only on the desired sub-space which represent a set of the *n* most preferred LO, where *n* is the number of the desired LO. If not, the resulting precision would be higher or equal to the real one, since the number of retrieved LO is less or equal to the number of desired LO at a given time instance.

Moreover, since the resulting LO space is restricted, the recall measure should also be measured over the space of the *n* most relevant LO, and not over the space of all relevant LO. This introduces the need for an alternative evaluation metric in adaptive content selection. In (Karampiperis and Sampson 2004), the *Selection Success* (SS) evaluation metric has been proposed as follows in (7.4):

$$\mathrm{SS}\,(\%) = 100 * \left(\frac{\#\,\textit{correct ranked LOs}}{\#\,\textit{requested LOs}} \right) \tag{7.4}$$

Although this metric seems similar to the precision metric (PM) in information retrieval systems, its difference is critical. It evaluates the precision of selecting LO not on the entire space of the Media Space, but only on the desired sub-space, and also takes into consideration the ranking of the selection process. This means that the proposed metric is stronger, since it measures the precision over a smaller value space.

7.4 Evaluation Methodology for Decision-Based AEHS

The underlying hypothesis of the design of a decision-based approach for content selection in AEHS is that it is feasible to construct a semi-automated algorithm, which generates a continuous decision function that estimates the desired AEHS response, aiming to overcome the above mentioned problems of insufficiency and inconsistency of the defined adaptation rule sets.

Thus, the goal of evaluating such an approach is twofold: first, to examine whether a proposed semi-automated decision based approach is capable of extracting decision models which replicate the AM of existing AEHS; and second, to verify via performance evaluation that this approach can be applied in cases where large-scale adaptation rule sets are needed to describe the desired AEHS response. To this end, the evaluation should be performed in two phases:

Phase A: Extracting the AM of existing AEHS. In this evaluation phase, the AM rules of existing AEHS are used for generating sample adaptation decisions. These decisions have the form of combinations of LO mapped to learner profiles, and are used to train the intelligent mechanism that fits the response function on these data.

The goal of this phase is to examine whether the proposed semi-automated decision based approach is capable of extracting the decision model of the AEHS in hand. In our experiments, we will try to extract the AM rules for content selection used in the INSPIRE (Papanikolaou et al. 2003) system.

Phase B: Scaling up the experiments. As already discussed, the problem of defining adaptation rules is a combinatorial problem, which means that in order to design sufficient and consistent adaptation rule sets, all the combinations of the adaptation decision variables should be covered. However, these combinations can be millions (Karampiperis and Sampson 2005), leading to huge rule sets that is difficult to author, manage and verify their sufficiency and/or consistency. To this end, in order to keep the adaptation rule set human-maintainable, existing AEHS in the literature use few adaptation variables, typically 2-4 variables for describing learners' behavior and 2-3 variables for describing educational content. The goal of this evaluation phase is to verify that the proposed approach can be applied in cases where large-scale adaptation rule sets are needed to describe the desired AEHS response. In order to do this, we simulate the existence of an AEHS that uses as many adaptation variables as possible. The variables learner profile properties and educational description model properties are selected from the items of wide-spread learning technology standards. However, special attention is given in generating learner profiles and educational content metadata records that simulate real-life conditions. Details on how such datasets are set are stated in Sect. 7.5.

7.5 Setting Up the Experiments

Before executing our experiments for measuring the performance of adaptive selection of LO, we need to design the media space and the learner model as the way explained in the next subsections.

7.5.1 Designing the Media Space

In the evaluation, we extract the AM of the INSPIRE system (Papanikolaou et al. 2003). INSPIRE system uses two variables in the educational resource description model, namely, the performance level and the learning resource type.

In the second evaluation phase, we simulate the existence of an AEHS where large-scale adaptation rule sets are needed to describe the desired AEHS response. To do so, we have used as educational resource description model a subset of the IEEE Learning Object Metadata (LOM) standard elements (IEEE 2002), illustrated in Table 7.1. The aggregation level and the relation/kind elements are used for structuring the media space.

Table 7.1 Educational resource description model used in evaluation phase B

IEEE LOM Category	IEEE LOM Element	Explanation
General	Structure	Underlying organizational structure of a LO
	Aggregation Level	The functional granularity of a LO
Educational	Interactivity Type	Predominant mode of learning supported by a LO
	Interactivity Level	The degree to which a learner can influence the aspect or behavior of a LO
	Semantic Density	The degree of conciseness of a LO
	Typical Age Range	Developmental age of the typical intended user
	Difficulty	How hard it is to work with or through a LO for the typical intended target audience
	Intended End User Role	Principal user(s) for which a LO was designed, most dominant first
	Context	The principal environment within which the learning and use of a LO is intended to take place
	Typical Learning Time	Typical time it takes to work with or through a LO for the typical intended target audience
	Learning Resource Type	Specific kind of LO. The most dominant kind shall be first
Relation	Kind	Nature of the relationship between two LO

In both evaluation phases, we need to simulate real-life conditions. This means that the simulated LO metadata records should have a distribution over their value spaces similar to the metadata value distribution found in real-life LO repositories.

(Najjar and Duval 2006) presented a statistical analysis of the actual use of IEEE LOM metadata elements in the ARIADNE LO repository. The results were derived from analyzing the empirical data (usage logs) of 3,700 ARIADNE metadata instances. Table 7.2 presents the percentage of times each ARIADNE data element was filled in by indexers during the indexing process.

Table 7.2 Usage percentage of data elements in ARIADNE repository

IEEE LOM Element	Value Provided (%)	Most used Vocabulary value (M)	% of M (filled-in)	%M among all cases
Aggregation Level	91.9	Lesson	92.7	85.2
Context	53.5	University Degree	69.7	37.2
Interactivity Level	53.2	Medium	67.7	36.1
Semantic Density	52.4	Medium	76.4	40.0
Difficulty Level	52.2	Medium	72.8	38.0
Restrictions	5.2	Contact Author	90	5.2
Source	1.3	-	-	-
Version Information	7.0	-	-	-
Description	11.2	-	-	-
OS Version	0.5	-	-	-
Installation Remarks	24.3	-	-	-
Other Constraints	0.15	-	-	-

From the data shown in Table 7.2, we notice that only one data element is almost always used: the aggregation level element. Other elements are used in about 50 % of the descriptions and the rest are rarely used in the indexing process. For the values of data elements, we can see that indexers often use just one value.

As a result, in order to simulate in our experiments the metadata of a real-world repository, we will generate metadata records with normal distribution over the metadata elements value space, simulating that not all metadata elements and their corresponding vocabulary terms are used equally. Normal distribution is a continuous probability distribution that is often used to describe random variables that tend to cluster around a single mean value.

7.5.2 Designing the Learner Model

In the first phase of the evaluation, we will extract the AM of the INSPIRE system (Papanikolaou et al. 2003). The INSPIRE system uses two variables in the learner model, namely, the learner's knowledge level and the learner's learning style.

In the second evaluation phase, we simulate the existence of an AEHS where large-scale adaptation rule sets are needed to describe the desired AEHS response. To do so, for the design of the learner model in our simulations, we have used an overlay model (Martins et al. 2008) for representing the learners' knowledge space and a stereotype model (Rich 1979) for representing learners' preferences. More precisely, for the learners' knowledge level we assume the existence of a related certification for each node of the learners' knowledge space, the evaluation score in testing records and the number of attempts made on the evaluation.

For modeling of learners' preferences we use learning styles according to (Honey and Mumford 1992), as well as modality preference information consisting of four modality types, namely, the visual modality, the textual modality, the auditory modality and any combination of the three modality preferences (Razmerita 2005). Each element of the learner model was mapped to the IMS Learner Information Package (LIP) specification (IMS 2001), as shown in Table 7.3.

In order to simulate in our experiments the profiles of real learners we generated profile records using truncated standard lognormal distribution with [sigma] = 1 and reduced by factor 1/5. This distribution is often used in the literature for simulating learner behavior (McCalla 2005).

7.5.3 *Simulating the AM of an AEHS*

The goal of our experiments is to evaluate the suitability of the set of performance evaluation metrics, presented in Sect. 7.3, for validating the use of decision-based approaches for adaptive LO selection in AEHS, and assess the use of these metrics in the case of our previously proposed statistical method for estimating the desired AEHS response.

Performance evaluation in this context means measuring how well our semi-automated approach fits the decision function to the provided model adaptation decisions (training data), and how well this decision function responds to decision cases not known during the training process (generalization capacity).

Table 7.3 Learner model used in evaluation phase B

Learner Model Element	IMS LIP Element	Explanation
Learning Style	Accessibility/ Preference/typename	The type of cognitive preference
	Accessibility/ Preference/prefcode	The coding assigned to the preference
Modality Preference	AccessForAll/ Context/Content	The type of modality preference
Knowledge Level	QCL/Level	The level/grade of the QCL
	Activity/ Evaluation/ noofattempts	The number of attempts made on the evaluation
	Activity/Evaluation / result/interpretscope	Information that describes the scoring data
	Activity/Evaluation/ result/score	The scoring data itself
Relation	Accessibility/ Preference/typename	The type of cognitive preference

As a result, we need to produce model adaptation decisions and compare them with the corresponding response of our decision-based approach. Some of these model adaptation decisions will be used for training our method, and some will be used for measuring its' generalization capacity.

In the first evaluation phase, the AM rules of an existing AEHS are used for generating sample adaptation decisions. In the second evaluation phase, we need to simulate the existence of an AEHS that uses as many adaptation variables as possible. Since such AEHS does not exist, we will simulate model adaptation decisions via the use of simulated instructional designers' preference models. These models have been selected in such a way that the preference surface is complex, thus, it would be a difficult task for the decision based algorithm to fit the training data.

To achieve this, we use as an instructional designers' preference model a multi-variable function, with 18 variables (k). These variables model the eleven (11) elements of the educational resource description model in use (that is, the elements used from the "general" and the "educational" IEEE LOM categories) and the seven elements of the learner model in use (Karampiperis and Sampson 2005). We assume that the response of this function expresses the utility of a given LO for a given learner profile (preference-based selection problem).

In our experiments, we simulate the preference models of five instructional designers, using multivariable non-convex functions. In our previous work (Karampiperis and Sampson 2004), we have defined the suitability/utility function of a learning object LO_i for the learner L_j as a function which varies from 0 to 1. This means that before we can use the non-convex functions as instructional designers' preference models, we need to scale them in the same value space. The normalisation equation that we use for this purpose is the (7.5):

$$F_{(f_{(x)})} = \frac{f_{(x)}^2}{1 + f_{(x)}^2} \tag{7.5}$$

where $f_{(x)}$ is the testing function. This formula, scales $f_{(x)} \in \Re$ to a new function $F_{(x)} \in [0,1)$, where $F_{(f_{(x)}=0)} = 0$, and $\lim_{f_{(x)} \to \pm\infty} F_{(f_{(x)})} = 1$.

For evaluating the performance, we have generated a set of 1.000 LO metadata records and a set of 100 learner profiles. In each experiment, 50% of the available LO metadata records, randomly selected, were used for algorithmic training and the rest 50% for measuring the generalisation, that is, the estimation capacity, of the algorithm. Similarly, in each experiment 50% randomly selected of the available learner profiles were used for algorithmic training and the rest 50% for measuring the generalisation of the algorithm.

7.6 Experimental Results and Discussion

We present experimental results from the execution of the above mentioned evaluation methodology for the case of our previously proposed statistical method for estimating the desired AEHS response (Karampiperis and Sampson 2005). The results are presented per evaluation phase.

7.6.1 Extracting the AM of Existing AEHS

Our first experiment was the application of our decision-based approach for replicating the AM of an existing AEHS. To this end, we simulated the AM of the INSPIRE (Papanikolaou et al. 2003), produced sample adaptation rules in the form of combinations of LO mapped to learner profiles, and applied our methodology to extract the AM. The INSPIRE system uses two variables from the learner model (namely, the learner's knowledge level and the learner's learning style) and two variables from the educational resource description model (namely, the performance level and the learning resource type), for performing adaptation decisions according to Table 7.4.

Fig. 7.1, presents the INSPIRE's AM dependencies of the learning style and learning resource type in the LO utility space, whereas Fig. 7.2 presents the same dependencies of the produced AM when our decision based approach is applied. From these figures we can observe that the produced AM is a super class of the INSPIRE's AM, since it contains more adaptation rules (dependencies between LO and learner characteristics). Moreover, we can observe that the produced AM has a continuous contour in the utility space, which means that this AM has the ability to always propose LO.

Table 7.4 INSPIRE AM rules (Papanikolaou et al. 2003)

Learner Attributes		Proposed LO	
Knowledge Level	Inadequate	Performance Level	Remember
	Mediocre		Use
	Advanced		Find
	Proficient		-
Learning Style	Activist	Learning Resource Type	Activity-oriented
	Reflector		Example-oriented
	Theorist		Theory-oriented
	Pragmatist		Exercise-oriented

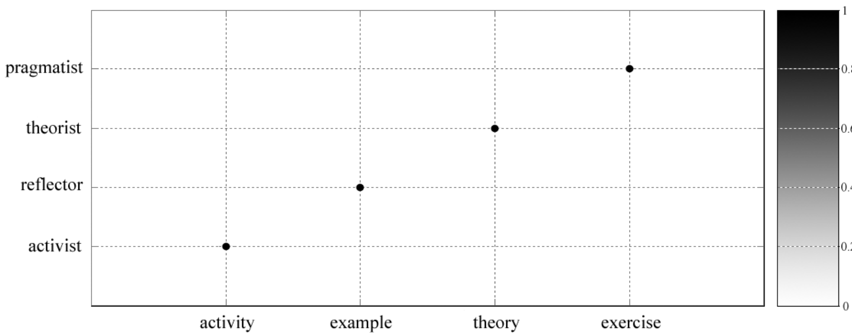


Fig. 7.1 INSPIRE: learning style and learning resource type utility space

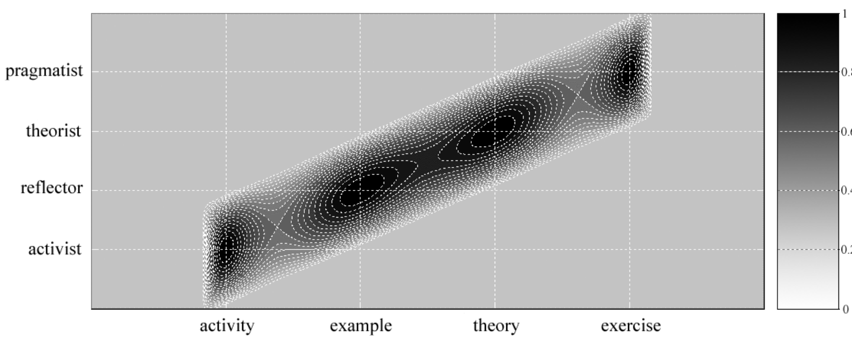


Fig. 7.2 Generated learning style and learning resource type utility space from INSPIRE

The designers of INSPIRE recognize as a problem when designing the INSPIRE system, the required effort for producing LO which cover all the combinations introduced by the INSPIRE AM Rules (Papanikolaou et al. 2003). This is due to the fact that the INSPIRE adaptation rules does not cover all the combinations of the free variables value space, e.g. what happens when a learner has knowledge level equal to “advanced” and learning style equal to “theorist”, but no theory-oriented LO with performance level equal to “find” exist in the LO repository. In this case, the INSPIRE system fails to provide a response, whereas by using our proposed decision based approach, the INSPIRE would respond with a suboptimal solution which would select the LO with the maximum utility for the given learner from the available ones.

After the above experiment, the research question was to investigate if the proposed decision based approach has the capacity of learning more complex AMs, consisting of many free variables (such as the adaptation variables presented in Table 7.1 and Table 7.3), with complex preference surfaces, thus, it would be a difficult task for the decision based algorithm to fit the training data. This is the goal of the second evaluation phase, which is presented in Sect. 7.6.2.

7.6.2 Scaling Up the Experiments

Before proceeding with the performance evaluation of our decision-based approach, we have conducted an additional experiment, aiming to assess the use of the performance evaluation metrics proposed by the literature.

Our semi-automated approach for adaptive content selection uses a preference-based LO selection mechanism based on the use of a suitability function that estimates the utility of a given LO for a given learner.

In order to compare the performance evaluation metrics discussed in Sect. 7.3, we evaluate the performance using randomly generated datasets which serve as model adaptation decisions and vary in size. The size of these datasets depends on the number of ranked LO for a given number of learner profiles. In real conditions, these rankings would be requested from an instructional designer. In our experiments, these rankings are the result of the application of the simulated instructional designers' preference models.

The datasets were divided into two subsets: the training dataset, which was used for algorithmic training and for evaluating the performance during the training process, and the generalization dataset, which was used for measuring the generalization capacity of the algorithm. Each experiment was executed 100 times using a randomly selected instructional designers' preference model.

Fig, 7.3 presents average selection performance results during algorithmic training, when using different simulation parameters regarding the number of learner profiles and the number of LO metadata records used. In each experiment, the selection performance was measured when using different values of the parameter n (varying from 10 to 500), which expresses the maximum number of requested LO from the Media Space. In this figure the performance evaluation was measured using the typical PM, the proposed alternative metric for SS, as well as, by applying the PM metric only on the desired sub-space of the media space (partial precision metric, PPM). From these results we observe the following:

1. Precision when measured with PM metric is independent from the maximum number of requested LO from the media space (selection space), as well as, from the ranking of the selected LO,
2. Precision when measured with PPM metric is independent from the ranking of the selected LO, but depends on the volume of the selection space,
3. The PPM metric tends to be equal to the PM metric when the selection space becomes bigger (n increases),
4. Performance evaluation using the PM metric is higher or equal to the performance when using the PPM metric. Also performance evaluation using the PM metric is higher or equal to the performance when using the SS metric,
5. The SS metric tents to be lower as the searching space increases, whereas PPM metric becomes higher as the searching space increases. This is due to the fact that, when the searching space increases the probability of introducing ranking errors also increases. Since the PPM metric is not dependent by the ranking of the selected LO, the PPM metric behaves differently from the SS metric.

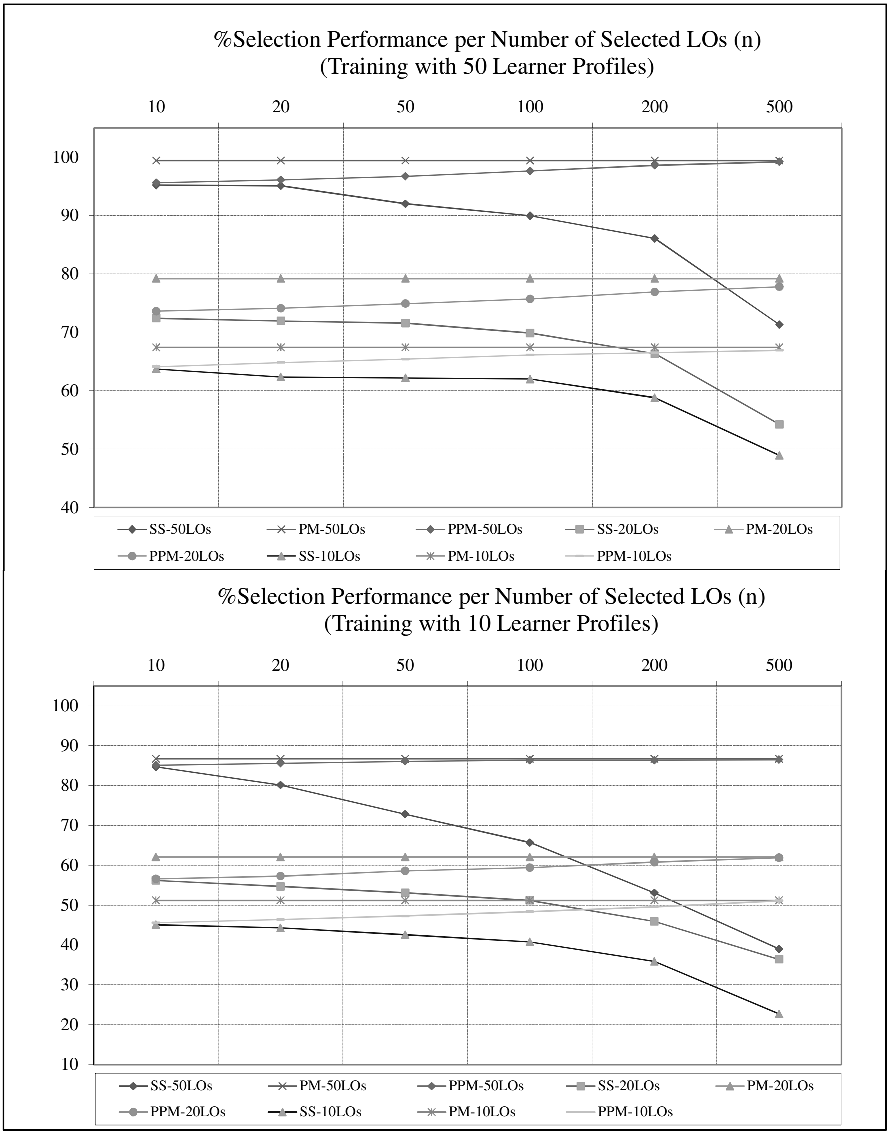


Fig. 7.3 Adaptive selection of LO - training results

The same observations apply also when measuring the generalization capacity, as depicted in Fig, 7.4. These observations verify the hypothesis that by definition the SS metric is stronger than the PM or the PPM metric. The PP and PPM metrics do not capture the precision errors resulting from ranking errors in the selected LO set.

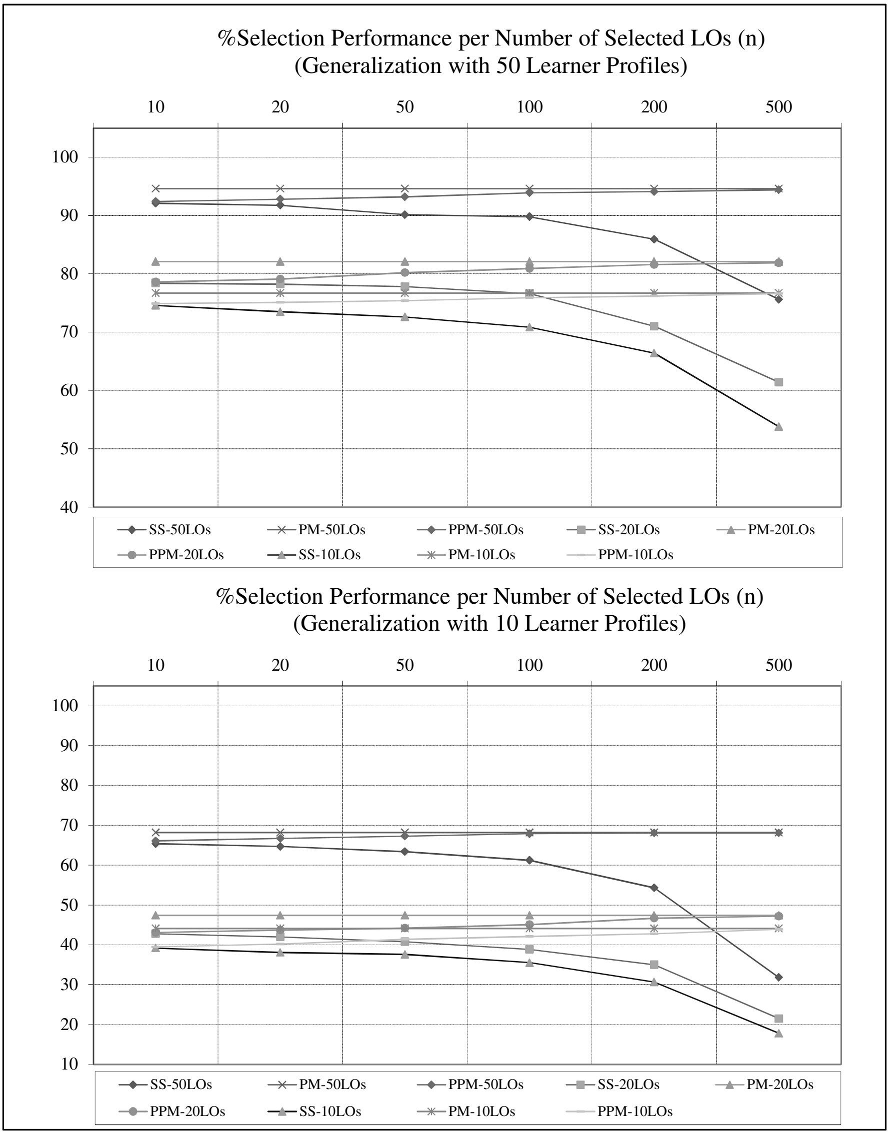


Fig. 7.4 Adaptive selection of LO - generalization results

In the case where the resulting LO sets hold the same LO but differ in the ordering of LO, the PP and PPM metrics remain the same (seam stable); whereas, the SS metric provides a realistic measurement of the precision. So in the case of AEHS, where the ranking of the selected LO is critical, the SS metric is used.

From these results we also observe that the SS depends on the requested LO from the media space (n), as well as the number of the LO and learner instances used for algorithmic training.

Additionally, for the same number of requested objects and the same number of learner profiles used, using more LO metadata records produces higher SS rates. Accordingly, for the same number of requested objects and the same number of used LO metadata records, using more learner profiles produces higher SS rates.

More analysis on the results presented in Fig, 7.3 and Fig, 7.4 show that, when the desired number of LO (n) is relatively small (less than 20), the selected LO by the decision model are close to those the instructional designer would select (with success rate over 70%), when using an input set consisting of more than 500 combinations of LO mapped to learner profiles (calculated as the multiplication of the LO with the learner profiles used).

By using the presented performance evaluation metrics, we can additionally investigate the influence of the explicit combinations required from the instructional designer (which are directly equivalent to the design effort required). To this end, we have executed additional experiments measuring the SS gain per number of requested combinations. This metric provides evidences about the trade-off that an instructional designer should make between the required effort and the improvement of the SS rate.

Fig, 7.5 presents simulation results of the design trade-off for combinations of LO metadata records with learner profiles that produce SS over the threshold of 70% for different values of the desired number of LO (n). From these results we observe that using a configuration of 500 combinations (which means classifying 50 LO metadata records over 10 learner profiles or vice versa) the gain in the SS rate is higher than using configurations with more combinations. The machine learning algorithm uses input knowledge in order to generate a continuous decision function that estimates the desired AEHS response. This knowledge comes in the form of combinations of LO mapped to learner profiles. When more input knowledge is provided, the machine learning algorithm fits better the response function on these data. However, there is a limit where this fitting process fails. If the algorithm is fed with too many input data, then it will over fit the response function over these data, losing its generalization capacity. Furthermore, we can observe that using the combination of 10 LO metadata records classified over 50 learner profiles leads to higher gain in the generalization success rate, whereas, using the opposite combination, that is, 50 LO metadata records classified over 10 learner profiles, leads to better results during the algorithmic training.

This is due to the fact that our decision based approach uses an interpolation method over the LO metadata space and an extrapolation mechanism over the learner profile space. This means that our approach learns from LO sequences associated with known learner profiles and generalizes its results to cover unknown learner profiles. Thus, using combinations with more LO leads to higher success rates during the training process, whereas, using combinations with more learner profiles leads to higher success rates during the generalization process. As a result, in order to minimize the required design effort and at the same time to maximize the SS rate, the combination of 10 LO metadata records classified over 50 learner profiles would be preferred.

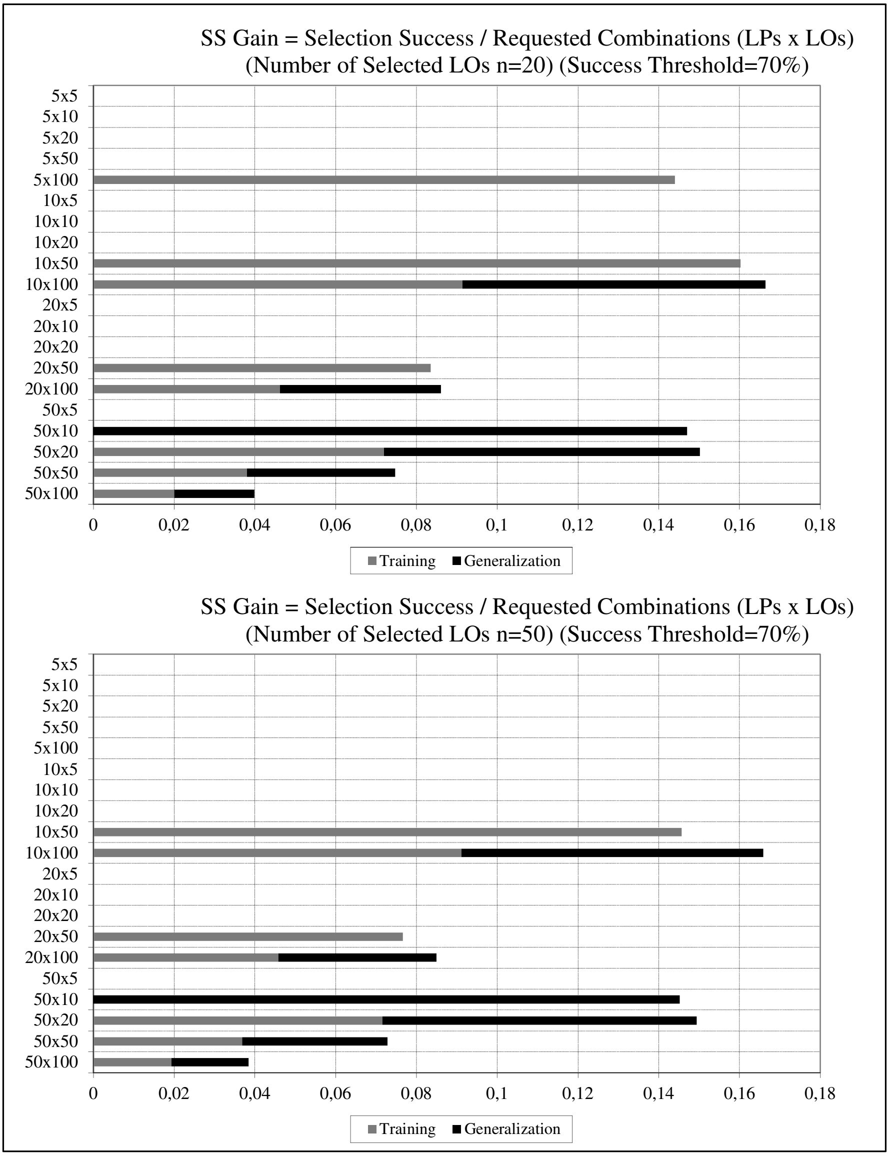


Fig. 7.5 Adaptive SS gain per requested input combinations

7.7 Summary and Future Research Directions

Adaptive LO selection is recognized as a challenging research issue in AEHS. In order to adaptively select LO in AEHS, the definition of adaptation behavior is required.

Several efforts have been reported in literature aiming to support the AM design by providing AEHS designers with either guidance for the direct definition of adaptation rules, or semi-automated mechanisms which generate the AM via the implicit definition of such rules.

The main drawback of the direct definition of adaptation rules is that there can be cases during the run-time execution of AEHS where no adaptation decision can be made. This is due to the fact that, even if appropriate resources exist in the media space, the absence of a required rule (insufficiency problem) or the conflict between two or more rules (inconsistency problem), prevents the AEHS to select and use them in the generated learning resource sequence. As a result, either less appropriate resources are used from the media space, or required concepts are not covered at all by the resulting sequence.

The goal of the semi-automated, decision-based approaches is to generate a continuous decision function that estimates the desired AEHS response, aiming to overcome the above mentioned problem. To achieve this, semi-automated approaches use data from the implicit definition of sample adaptation rules and attempt to fit the response function on these data. Although such approaches bare the potential to provide efficient AMs, they still miss a commonly accepted framework for evaluating their performance.

In this chapter, we discussed a set of performance evaluation metrics that have been proposed by the literature for validating the use of decision-based approaches in adaptive LO selection, and assessed the use of these metrics in the case of our previously proposed statistical method for estimating the desired AEHS response.

More precisely, we discussed the limitations of the performance metrics used by the literature for the problem of adaptive content selection, introduced the need for an alternative evaluation metric and presented a metric, which although seems similar to the PM in information retrieval systems, its difference is critical. This metric evaluates the precision of selecting LO not on the entire space of the media space, but only on the desired sub-space, and also it takes into consideration the ranking of the selection process.

Future research includes the study of variations of the presented performance evaluation metrics, as well as, the investigation of a comparison metric between rule-based and decision based AEHS. In the context of AEHS an interesting research question is the separation of the learning scenario from the AM. By this way, we anticipate, on one hand, to support the sequencing of unstructured raw media, and on the other hand, to facilitate the support of different pedagogical strategies without redesigning the AM rules. Moreover, the decomposition of LO from existing courses, allowing reuse of the disaggregated LO in different educational contexts is considered as an important research question.

The intelligent selection of the disaggregation level and the automatic structuring of the atoms (raw media) inside the disaggregated components in order to preserve the educational characteristics they were initially designed for, is a key issue in the research agenda for LO (Duval and Hodgins 2003).

References

Alfonseca, E., Rodriguez, P., Perez, D.: An approach for automatic generation of adaptive hypermedia in education with multilingual knowledge discovery techniques. Computers & Education 49, 495–513 (2007)

Aroyo, L., Mizoguchi, R.: Towards evolutional authoring support systems. Journal of Interactive Learning Research 15(4), 365–387 (2004)

Aroyo, L., Mizoguchi, R., Tzolov, C.: OntoAIMS: Ontological approach to courseware authoring. In: Proceedings of ICCE, pp. 1011–1014 (2003)

Biletskiy, Y., Baghi, H., Keleberda, I., Fleming, M.: An adjustable personalization of search and delivery of learning objects to learners. Expert Systems with Applications 36(5), 9113–9120 (2009)

Brusilovsky, P.: Adaptive Navigation Support. In: Brusilovsky, P., Kobsa, A., Nejdl, W. (eds.) Adaptive Web 2007. LNCS, vol. 4321, pp. 263–290. Springer, Heidelberg (2007)

Brusilovsky, P., Henze, N.: Open Corpus Adaptive Educational Hypermedia. In: Brusilovsky, P., Kobsa, A., Nejdl, W. (eds.) Adaptive Web 2007. LNCS, vol. 4321, pp. 671–696. Springer, Heidelberg (2007)

Brusilovsky, P., Wade, V., Conlan, O.: From learning objects to adaptive content services for E-learning. In: Pahl, C. (ed.) Architecture Solutions for E-Learning Systems, pp. 243–261. Idea Group Inc. (2007)

Conlan, O., Hockemeyer, C., Wade, V., Albert, D.: Metadata driven approaches to facilitate adaptivity in personalized eLearning systems. Journal of the Japanese Society for Information and Systems in Education 1(1), 38–44 (2002)

Cristea, A.I.: Authoring of adaptive educational hypermedia. In: Proceedings of ICALT, pp. 943–944 (2007)

Cristea, A.I., Kinshuk: Consideration on LAOS, LAG and their Integration in MOT. In: Proceedings of ED-MEDIA, pp. 511–518 (2003)

Dagger, D., Wade, V., Conlan, O.: Personalisation for all: Making adaptive course composition easy. Education Technology & Society 8(3), 9–25 (2005)

De Bra, P.: Web-based educational hypermedia. In: Romero, C. (ed.) Data Mining in E-Learning, pp. 3–17. WIT Press (2006)

De Bra, P., Houben, G., Wu, H.: AHAM: A Dexter based reference model for adaptive hypermedia. In: Proceedings of HT, pp. 147–156 (1999)

Dolog, P., Henze, N., Nejdl, W., Sintek, M.: The Personal Reader: Personalizing and Enriching Learning Resources Using Semantic Web Technologies. In: De Bra, P., Nejdl, W. (eds.) AH 2004. LNCS, vol. 3137, pp. 85–94. Springer, Heidelberg (2004)

Dolog, P., Simon, B., Nejdl, W., Klobucar, T.: Personalizing access to learning networks. ACM Transactions on Internet Technology 8(2), Article 8 (2008)

Duval, E., Hodgins, W.: A LOM research agenda. In: Proceedings of WWW (2003)

Harman, K., Koohang, A.: Learning objects: standards, metadata, repositories & LCMS. Informing Science Press, California (2006)

Henze, N., Nejdl, W.: A logical characterization of adaptive educational hypermedia. New Review of Hypermedia and Multimedia 10(1), 77–113 (2004)

Honey, P., Mumford, A.: The manual of learning styles. Maidenhead (1992)

Hsieh, T.C., Chiu, T.K., Wang, T.I.: An approach for constructing suitable learning path for documents occasionally collected from internet. In: Proceedings of ICMLC, pp. 2138–3143 (2008)

Huang, M.J., Huang, H.S., Chen, M.Y.: Constructing a personalized e-learning system based on genetic algorithm and case-based reasoning approach. Expert Systems with Applications 33, 551–564 (2007)

Huang, Y., Chen, J., Huang, T., Jeng, Y., Kuo, Y.: Standardized course generation process using dynamic fuzzy Petri nets. Expert Systems with Applications 34, 72–86 (2008)

IEEE. IEEE learning object metadata standard (IEEE LOM). Technical specification. IEEE (2002)

IMS. Learner information package (LIP). Technical specification. IMS Global Learning Consortium Inc. (2001)

Karampiperis, P., Sampson, D.G.: Adaptive learning object selection in intelligent learning systems. Journal of Interactive Learning Research 15(4), 389–409 (2004)

Karampiperis, P., Sampson, D.G.: Adaptive learning resources sequencing in educational hypermedia systems. Educational Technology & Society 8(4), 128–147 (2005)

Kiu, C.C., Lee, C.S.: Learning objects reusability and retrieval through ontological sharing: A hybrid unsupervised data mining approach. In: Proceedings of ICALT, pp. 548–550 (2007)

Knutov, E., De Bra, P., Pechenizkiy, M.: AH 12 years later: a comprehensive survey of adaptive hypermedia methods and techniques. New Review of Hypermedia and Multimedia 15(1), 5–38 (2009)

Kravčík, M., Specht, M.: Flexible Navigation Support in the WINDS Learning Environment for Architecture and Design. In: De Bra, P., Nejdl, W. (eds.) AH 2004. LNCS, vol. 3137, pp. 156–165. Springer, Heidelberg (2004)

Lee, M., Tsai, K., Wang, T.: An ontological approach for semantic-aware learning object retrieval. In: Proceedings of ICALT, pp. 208–210 (2006)

Martins, A.C., Faria, L., Vaz de Carvalho, C., Carrapatoso, E.: User modeling in adaptive hypermedia educational systems. Educational Technology & Society 11(1), 194–207 (2008)

McCalla, G.: Smart recommendation for an evolving e-learning system: architecture and experiment. International Journal on E-Learning 4(1), 105–129 (2005)

McGreal, R.: Online education using learning objects. Routledge, London (2004)

Najjar, J., Duval, E.: Actual use of learning objects and metadata: An empirical analysis. IEEE Technical Committee on Digital Libraries Bulletin 2(2) (2006)

Nejdl, W., Brusilovsky, P.: Adaptive hypermedia and adaptive web. In: Singh, M. (ed.) Practical Handbook of Internet Computing, pp. 1.1–1.12. CRC Press (2004)

Ochoa, X., Duval, E.: Relevance ranking metrics for learning objects. IEEE Transactions on Learning Technologies 1(1), 34–48 (2008)

Papanikolaou, K., Grigoriadou, M., Kornilakis, H., Magoulas, G.: Personalising the interaction in a web-based educational hypermedia system: The case of INSPIRE. International Journal of User Modeling and User-Adapted Interaction 13(3), 213–267 (2003)

Ras, E., Ilin, D.: Using Decision Models for the Adaptive Generation of Learning Spaces. In: Nejdl, W., Kay, J., Pu, P., Herder, E. (eds.) AH 2008. LNCS, vol. 5149, pp. 153–162. Springer, Heidelberg (2008)

Razmerita, L.: User modeling and personalization. In: Sherry, Y.C., Magoulas, G.D. (eds.) Adaptable and Adaptive Hypermedia Systems, Idea Group Inc. (2005)

Rich, E.: User modeling via stereotypes. Cognitive Science 3(4), 329–354 (1979)

Sampson, D., Karampiperis, P.: Decision models in the design of adaptive educational hypermedia systems. In: Graf, S., Lin, F., Kinshuk, McGreal, R. (eds.) Intelligent and Adaptive Learning Systems: Technology Enhanced Support for Learners and Teachers. Idea Group Inc. (2011)

Stash, N., Cristea, A.I., De Bra, P.: Adaptation languages as vehicles of explicit intelligence in adaptive hypermedia. International Journal of Continuing Engineering Education and Lifelong Learning 17(4-5), 319–336 (2007)

Wang, T.I., Tsai, K.H., Lee, M.C., Chiu, T.K.: Personalized learning objects recommendation based on the semantic-aware discovery and the learner preference pattern. Educational Technology & Society 10(3), 84–105 (2007)

Wu, H., De Bra, P.: Sufficient Conditions for Well-Behaved Adaptive Hypermedia Systems. In: Zhong, N., Yao, Y., Ohsuga, S., Liu, J. (eds.) WI 2001. LNCS (LNAI), vol. 2198, pp. 148–152. Springer, Heidelberg (2001)

Abbreviations

AEHS	Adaptive Educational Hypermedia Systems
AHA	Adaptive Hypermedia Architecture
AHAM	Adaptive Hypermedia Application Model
AM	Adaptation Model
ATO	Authoring Task Ontology
ICPC	International Classification of Primary Care
LIP	Learner Information Package
LO	Learning Object
LOM	Learning Object Metadata
MeSH	Medical Subject Headings
MOT	My Online Teacher
PM	Precision Metric
PPM	Partial Precision Metric
SS	Selection Success

Chapter 8
PCMAT – Mathematics Collaborative Educational System

Constantino Martins[1], Luiz Faria[1], Marta Fernandes[1], Paulo Couto[1], Cristina Bastos[1], and Eurico Carrapatoso[2]

[1] GECAD – Knowledge Engineering and Decision Support Group / Institute of Engineering – Polytechnic of Porto
R. Dr. António Bernardino de Almeida, 4200-072 Porto, Portugal
{acm,lef,mmaf,pjco}@isep.ipp.pt, cristinabastos70@gmail.com

[2] Faculty of Engineering of the University of Porto
R. Dr. Roberto Frias 4200-465 Porto, Portugal
emc@fe.up.pt

Abstract. PCMAT (Mathematics Collaborative Learning Platform) is a collaborative adaptive learning tool based on progressive assessment for Mathematics in Basic Schools. The learning platform is based on a constructivist approach, assessing the user knowledge and presenting contents and activities adapted to the characteristics and learning style of the student. The intelligent behavior of such platform depends on the existence of a tentative description of the student – the student model (SM). The contents of this model and the student most prominent learning style are used by a domain and interaction model to select the most appropriate response to student actions. The SM is used to select the more appropriate learning object according to the student learning stage. However, this selection demands for the access to metadata describing the contents of the learning object. This need leaded to the application of a standard to describe the learning objects (LO). The authoring of LO corresponds to one major maintenance costs present in these applications. PCMAT is able to generate several instances of the same learning object through the parameterization of some features of the learning object. The platform is also able to process student responses in natural language. This project shows how techniques from Adaptive Hypermedia System (AHS) field can improve e-learning based systems in a basic school environment.

8.1 Introduction

The main goal of Educational Adaptive Systems (EAS) is to achieve applications and also able to adequate its relation with the student in terms of content presentation, navigation, and interface according to a predefined but updatable SM (Brusilovsky 2001, De Bra et al. 2004).

A. Peña-Ayala (Ed.): Intelligent and Adaptive ELS, SIST 17, pp. 183–212.
springerlink.com

In this kind of systems, the emphasis is placed on the student knowledge about the domain application and his learning style, in order to allow him to reach the learning objectives proposed in his training (Chepegin et al. 2004).

Although numerous research and already developed systems have provided good results, more development, experimentation and implementation are still necessary to conclude about the adequate features and effectiveness of these systems (Martins et al. 2008a, Martins et al. 2008b).

The aims of this document are: to define what is a SM, to present existing and well known SM, to compare existing intelligent systems in the scientific area of student modeling and to present the project: PCMAT.

PCMAT is a collaborative learning platform based on a constructivist approach, assessing the user knowledge and presenting contents and activities adapted to the characteristics and learning style of the student of mathematics in basic schools.

This chapter is organized as follows. The first three sessions present a brief state of the art concerning the major concepts involved in an EAS. In particular, Sect. 8.2 provides a general approach to AHS, Sect. 8.3 defines SM and its role in the adaptation process, and Sect. 8.4 deals about the learning styles concept. In Sect. 8.5 some standards used to describe metadata about LO are described. Sect. 8.6 presents some issues regarding PCMAT implementation. Finally, Sect. 8.7, and 8.8 present some results and conclusions.

8.2 Adaptive Hypermedia Systems

AHS term is generally referred as a crossroad in the re-search of Hypermedia and User Model (UM) (De Bra et al. 2004, Brusilovsy, 1996, Brusilovsky, 2001). An AHS builds a model of the objectives, preferences and knowledge of each user and uses it, dynamically, through the Domain Model and the Interaction Model, to adapt its contents, navigation and interface to the user needs.

(Chepegin et al. 2004) indicate: "These systems must present the functionality to change content presentation, links structure or links annotation with the following objectives: guiding the user to relevant information and keep him away from the irrelevant one, or pages that he still would not be able to understand". This objective is generally known as link adaptation; supplying, in the content (page), additional or alternative information to certify that the most relevant information is shown. It is generally known as content adaptation.

The global architecture proposed by (Benyon 1993) and (De Bra et al. 2004) indicates that AHS must have three essential parts:

- The UM that describes the information, knowledge, and preferences of the user. This component allows extracting and expressing conclusions on the user characteristics,

- The Domain Model represents a set of domains concepts; in different AHS these concepts can have distinct functions, weights, and meanings; most commonly, each concept is connected/related with other concepts, representing a semantic net,
- The Interaction Model, which represents and defines the interaction between the user and the application.

In educational adaptive hypermedia, the emphasis is placed on students' knowledge in the domain application and learning style, in order to allow them to reach the learning objectives proposed in their training (Martins et al. 2008b).

8.3 Student Model

The beginning of user modeling is dated to 1978/1979 with the first work by Allen, Cohen, Perrault, and Rich (Kobsa 1993). In the next 10 years, numerous applications or systems were developed to store different types of user information to allow distinct adaptation models. Morik, Kobsa, Wahlster, and McTear present an extensive survey of these systems (Kobsa 1993). In these initial systems, user modeling was embedded and there was not a clear distinction from other components of the system (Kobsa 1993).

In 1990, Kobsa was the first author to use the term "User Modeling Shell System". Since then, different systems have been developed with the ability to reuse UM (Kobsa 1993, Martins et al. 2008b).

In generic AHS, the UM allows changing several aspects of the system, in reply to certain characteristics (given or inferred) of the user (Brusilovsky 2001). These characteristics represent the knowledge and preferences that the system assumes the user (individual, group of users or no human user) has (Martins et al. 2008a, Martins et al. 2008b).

In educational adaptive hypermedia systems, the UM (or SM) has increased relevance: when the student reaches the objectives of the course, the system must be able to re-adapt, for example, to his knowledge (Brusilovsky 2001, Martins et al. 2008a, Martins et al. 2008b).

A SM includes information referring to the specific knowledge that the system judges that the user possesses on the domain, known as the Domain Dependent Data (DDD). The components of the DDD correspond to the Domain Model with three-level functionality (Benyon 1993):

1. Task level: with the objectives/competences of the domain that the user will have to master. In this case, the objectives or intermediate objectives can be altered according to the evolution of the learning process,
2. Logical Level: which describes the user knowledge of the domain and is updated during the students learning process,
3. Physical Level: it registers and infers the profile of the user knowledge.

The Domain Independent Data (DID) is composed of two elements: the Psychological Model and the Generic Model of the Student Profile, with an explicit representation (Kobsa 1993). The psychological data are related with the cognitive and affective aspects of the student. Some studies have demonstrated that the difference between the cognitive capacities and personality aspects affects the quality of some models or styles of interaction (Kobsa 1993). This data are more permanent which allows the system to know beforehand that it must adapt to which the characteristics are (Benyon 1993, Vassileva 1996).

The data related to the user interests, common knowledge and background is kept in the Generic Model of the Student Profile. The DID includes the following aspects (Benyon 1993, Kobsa 1993, Martins et al. 2008b): Initial user knowledge, Objective and plans, Cognitive capacities, Learning styles, Preferences, Academic profile (technological studies versus economical studies and management, knowledge of literature, artistic capacities, etc.), Age and type of student, Cognitive style (affective, impulsive, etc.), and personality aspects (introverted, extroverted, etc.).

As expressed before, some of these characteristics are relevant for a determined type of UM and not for others (Brusilovsky 1996, Brusilovsy 2001, Martins et al. 2008b). Therefore, for each AHS it will be necessary to define the characteristics and relevant parameters of the user to be kept (Martins et al. 2008a).

The following list tries to address the most common aspects that support adaptation (Martins et al. 2008b):

- DID:
  - Generic profile: Personal information (name, email, password, etc.), demographic data (age, etc.), academic background, qualifications, background knowledge, deficiencies: visual or others, the Domain of Application, and inheritance of characteristics (creation of user stereotypes),
  - Psychological profile: Learning style (taxonomy), cognitive capacities, personality; inheritance of characteristics.
- DDD: objectives, plan, complete description of the navigation, knowledge acquired, results of evaluations, context model, aptitude; interests (definition of the interests of the individual with the objective to adapt the navigation and contents), and deadline extend (long, short, or normal stated period).

Two different types of techniques are used to implement the SM: knowledge and behavioral based (Martins et al. 2008b). The knowledge-based adaptation typically results for data collected through questionnaires and studies of the user, with the purpose to produce a set of initial heuristics. The behavioral adaptation results from the monitorization of the user during his activity.

The use of stereotypes classifies users in groups and generalizes student characteristics to that group (Martins et al. 2008b). The definition of the necessary characteristics for the classification in stereotypes must take to consideration the granularity degree wanted.

The behavioral adaptation can be implemented in two forms: the Overlay and the Perturbation methods (Martins et al. 2008b).

Those methods relate the student's knowledge level with the learning objectives that he intends to reach (Martins et al. 2008b). Table 8.1 represents some characteristics present in AHS UM. Further, some systems based on the overlay model will be described. Some AHS that use the overlay model for UM are the next:

- The Adaptive Hypermedia Architecture System (AHA) is an Educational AHS. The purpose of this system is to deliver courses over the web. The UM is based on concepts knowledge that the user acquires by solving tests and reading the hypermedia pages of the course,
- The XAHM system in which the adaptation depends on the users level of expertise about the known concepts of the system domain (which are a subset of all domain concepts),
- The ISIS-TUTOR is a system intended for learning the print formatting language of an information retrieval system CDS/ISIS/M which uses the overlay model with a set of integer counters,
- The HYPERFLEX, which is an adaptive hypertext browser. This system asks the user to specify his objectives and plans and uses a connected semantic network (Brusilovsky 1996).

Table 8.1 Some UM characteristics of some existing AHS

Systems \ Characteristics	User knowledge	Stereotypes	User objectives	Prerequisite and expertise	Preferences	User interests	History
ADAPTWEB				X		X	X
AHA	X		X	X		X	
AVANTI	X	X	X				
C-BOOK		X					
ANATOM-TUTOR	X	X					
ELM-ART	X		X			X	X
INTERBOOK	X		X	X	X	X	X
KBS HYPERBOOK	X		X	X	X	X	X
INSPIRE	X		X			X	
HYPADAPTER	X				X		
HYPERFLEX	X		X		X		
HYPLAN			X				
HYNECOS		X		X	X		X
ISIS-TUTOR	X						
KN-AHS	X						
METADOC	X	X					
XAHM	X		X		X	X	

Many systems use stereotypes for describing the user. HYPERTUTOR is a system that uses stereotypes for describing the user. This system employs exercises to obtain information about the users and uses stereotypes for UM. The student can belong to one of three groups: novice, medium or expert (Kavcic 2000).

Many times one method alone does not allow the modeling needs of the system and the combination of diverse methods has to be chosen (Martins et al. 2008b).

8.4 Learning Styles

The key of constructivism theory is that the student must be actively involved in the learning process. It is important that teachers understand that the construction of knowledge acquisition occurs from knowledge that the student already possesses and differs from student to student. The role of the teachers is now to be a guide for the student (Jonassen 1991, Martins et al. 2008a). Students learn in different ways and depend upon many different and personal factors (Kolb 2005).

The emphasis in students individual differences is also important in a context to recognize, design and support students activities (tasks). In constructivism learning theory, students have different learning styles. Also, the capacity of adaptation in different social contexts and the constructive social aspect of knowledge must be taken in consideration (Jonassen 1991, Martins et al. 2008a).

Generally, learning styles is understood as something that intent to define models of how a person learns. The application of strategies compatible with the preferred learning style usually leads to better results. Some case studies have been proposed that teachers should assess the learning styles of their students and adapt their classroom and methods to best fit each student's learning style (Kolb 2005, Stash et al. 2005). There are different learning styles models (based on different psychological theories) such as for example models based on (Kolb 2005):

- Personality (Ritu and Sugata 1999),
- Information processing approach (Schmecks 1983),
- Social Interaction (Reichmann and Grasha 1974),
- Multidimensional factors (Ritu and Sugata 1999).

VARK Strategies is a questionnaire that provides users with a profile of their learning preferences. These preferences are about the ways that they want to access and select information. These models/strategies describe three basic learning styles: visual learning (learn by seeing), auditory learning (learn by reading or hearing), and kinesthetic learning (learn by doing) (Martins et al. 2008a).

The model proposed by Kolb is the most commonly used inventory and is based on Piaget's model on cognitive and learning development (Kolb 2005). Kolb Learning Styles Model is based on the four stages of the learning cycle: Concrete Experience (CE), Reflective Observation (RO), Abstract Conceptualization (AC) and Active Experimentation (AE) (Kolb 2005, Stash et al. 2005).

From these levels are defined the matrix to allow the classification of the Student Learning Styles (Table 8.2). The learning process must take into consideration the individual cognitive and emotional parts of the student. Student personal progress must be adapted and not generalized and repetitive (Jonassen 1991, Martins et al. 2008b).

8.5 Metadata Standards

Inherent to the operation of PCMAT is the existence of a collection of digital LO and the need for these to be retrieved manually, to be managed by teachers and content developers, and automatically, to be presented to the students in accordance with the respective learning style and knowledge. To make this possible, a metadata record will be associated to each learning object. This metadata document will contain the information pertaining to the learning object creators, the learning object identification, such as title, short description and keywords to help the search and retrieval actions, and also information pertaining to the learning object pedagogical characteristics, to allow for the learning object to be retrieved by the system if found suitable to a particular students knowledge and learning style.

With these requirements in mind we were confronted with the option of developing a PCMAT specific set of metadata or adopting an established standard. To be adopted, it was required that the metadata standard: 1) was extendable and admitted the inclusion of alien elements; 2) admitted an eXtensible Markup Language (XML) document as a representation of a metadata instance, thus providing the necessary XML Schema.

As far as LO are concerned, there are two established metadata schemas currently in use: the IEEE Learning Object Metadata (LOM) and the Dublin Core Metadata Element Set (DCMES) (Barker 2010, Currier 2008).

The IEEE LOM is a multi-part standard, currently consisting of a conceptual data schema (IEEE 2002) and its XML schema binding (IEEE 2005). This standard defines a structured set of 76 data elements, covering a wide variety of characteristics found to be relevant to define a learning object, and grouped in the following categories (illustrated in Fig. 8.1):

Table 8.2 Kolb learning styles matrix (Kolb 2005)

	Doing Active Experimentation (AE)	Watching Reflective Observation (RO)
Feeling Concrete Experience (CE)	Accommodating (CE/AE)	Diverging (CE/RO)
Thinking Abstract Conceptualization (AC)	Converging (AC/AE)	Assimilating (AC/RO)

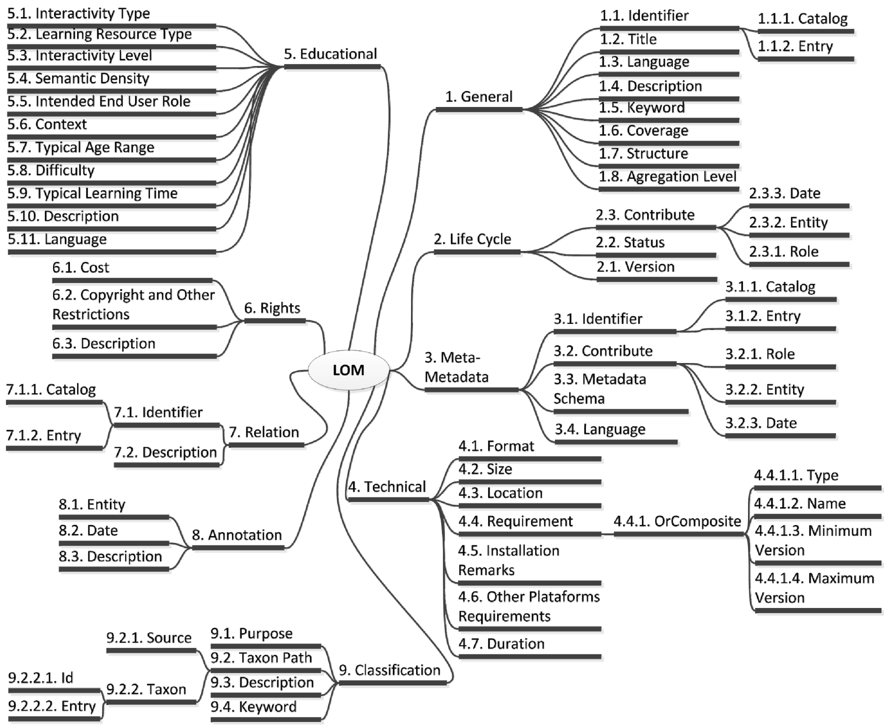


Fig. 8.1 Elements and structure of the LOM conceptual data schema (from IMS Meta-data Best Practice Guide for IEEE 1484.12.1-2002 Standard for Learning Object Metadata, Version 1.3 Final Specification. http://www.imsglobal.org/metadata/mdv1p3/ immdbestv1p3.html)

- General information that describes the LO as a whole, as, for example, an identifier, the title, a description, and a set of keywords,
- Life cycle information pertaining to the development of the learning object,
- Meta-metadata information concerning the actual metadata document and not the described learning object,
- Technical information regarding technical requirements and technical characteristics of the learning object,
- Educational information about the LO educational and pedagogic aspects,
- Rights information on the LO intellectual property rights and conditions of use,
- Relation information that defines the relationship of the described learning object with other learning objects,
- Annotation space for storing comments on the learning objects usage,
- Classification description of the learning object in accordance with different classification systems.

Within each category an element or a group of elements may be repeated, if necessary, and every element is optional. According to the standard, a LOM instance that contains no value for any of the LOM data elements is a conforming instance (IEEE 2002).

Nonetheless, if use is made of a LOM data element, this shall be made respecting its structure within the schema and its value should be in accordance with the data types and vocabularies defined in the schema. However, the LOM standard allows the insertion of elements, or even attributes to the LOM elements, other than the ones foreseen within the standard, provided that these are identified by a specific namespace. This possibility allows a community of users to specify which elements and vocabularies they will use, building a LOM application profile.

Perhaps due to its status as an international standard, or because it was developed from the beginning with the purpose of characterizing a learning object, or maybe due to the possibility of developing specific application profiles, LOM has been widely implemented by repositories. ARIADNE, SMETE, Learning Matrix, iLumina, MERLOT, HEAL, CAREO, Learn Alberta Online Curriculum Repository and Lydia Inc. are some examples of repositories that implemented the LOM standard (Neven 2002).

The DCMES is a vocabulary of fifteen properties for use in resource description (DCMI 2010). Standardized as ISO Standard 15836:2009, the core Element Set is intended to be broad and generic, usable for describing a wide range of resources and comprises the following elements: contributor, coverage, creator, date, description, format, identifier, language, publisher, relation, rights, source, subject, title and type. All these elements are optional and may be repeated if required. Dated 1998, this core set in now part of a larger set of metadata vocabularies known as DCMI Meta-data Terms (DCMI-TERMS). Aware of the fact that the metadata needs of particular communities and applications are very diverse, the Dublin Core Metadata Initiative (DCMI) provides a framework for designing a Dublin Core Application Profile (DCAP). As stated in the Guidelines for DCAP (DCMI 2008), a DCAP is a generic construct for designing metadata records that does not require the use of DCMI-TERMS, a DCAP can use any terms that are defined on the basis of Resource Description Framework (RDF), combining terms from multiple namespaces as needed.

A DCAP is a document (or set of documents) that specifies and describes the metadata used in a particular application, including guidance for metadata creators and clear specifications for metadata developers, and it consists of the following components:

- Functional requirements (mandatory) describes what a community wants to accomplish with its application,
- Domain model (mandatory) characterizes the types of things described by the metadata and their relationships,
- Description set profile (mandatory) enumerates the metadata terms to be used and the rules for their use,

- Usage guidelines (optional) describe how to apply the application profile, how the used properties are intended to be used in the application context,
- Syntax guidelines (optional) define the machine syntax that will be used to encode the data.

Still in accordance with the aforementioned guidelines, application profiles should be developed by a team with specialized knowledge of the resources that need to be described, the metadata to be used in the description of those resources, as well as an understanding of the Semantic Web and the linked data environment. Thus, following this recommendation, the DC-Education Community (DC-Ed) decided at the DC-2007 Conference in Singapore to form a new Task Group in order to finish the DC-Education Application Profile by mid-2008 (DCMI 2007). However, in the meeting held by the DC-Ed on October 20th, 2010, in Pittsburgh, the development work was still being carried out on the domain model of the DC-Education Application Profile. At this meeting it was foreseen that the complete documentation would be available since July-August 2011 (DCMI 2007).

Notwithstanding, DCMES has been adopted in particular by libraries and archives worldwide, mostly because it is an integral part of the Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH) (OAI 2008). The OAI-PMH allows for compliant metadata from different repositories to be harvested automatically in order to build a centralized point of search. The implementation of this protocol has been quite successful due to the availability of open source freeware like DSpace, Fedora or Greenstone.

In the view of characteristics and development of the IEEE LOM and DCMES, and since PCMAT is at an early stage, it was decided to adopt IEEE LOM as the basis for the development of PCMATs domain model, mainly because it defines a large set of metadata from which one can choose the elements found relevant to characterize a specific set of LO and also because it allows the insertion of non-LOM elements or attributes if deemed necessary.

8.6 PCMAT Platform

The PCMAT system allows students to autonomously create and consolidate knowledge, with permanent automatic feedback and support, through instructional methodologies and educational activities explored in a constructivist manner.

The adaptation of the application is based on progressive self-assessment exercises solved by the student that evolve in difficulty and domain topics. The curriculum is defined by the teacher but is dynamically individualized to the student according to the current level of knowledge, competences, abilities, and learning path. The platform provides contextualized access to tutorials if the students fail a progression step.

Others goals of the PCMAT include:

1. To define a new strategies and architecture for the implementation of an Adaptive Hypermedia Educational platform to support and improve Mathematics in basic schools,
2. To define the student model that describes the information, knowledge, preferences, and learning style of the user (Sect. 8.6.2),
3. To define the process and tools to produce LO aligned with the LOM standard, and conceive a knowledge representation of the students attributes concerning the emotional characteristics and develop a set of adaptive and dynamic pedagogical strategies to put forward this hybrid model (Sect. 8.6.3);
4. To develop the system functionalities for the interaction model (interface adaptation) considering the objectives, profile and knowledge of the student and the Domain and Adaptation model (Sect. 8.6.4),
5. To improve results and knowledge in Mathematics in Basic Schools (Sect. 8.7 and 8.8).

8.6.1 System Architecture

The PCMAT platform application is based on AHA. AHA is a Web based adaptive hypermedia system and is able to perform adaptation that is based on the user's browsing actions (De Bra et al. 2004). However, our system has significant differences with AHA, namely:

- The definition and implementation of the SM (Sect 8.6.2),
- The existence of an authoring tool for metadata for the LO (Sect. 8.6.3),
- The definition and implementation of the domain model and adaptation rules,
- The application for the creation of questions and automatic generation of tests (Sect. 8.6.4),
- The pedagogical model definition and implementation (Sect. 8.6.4).

PCMAT is a project built on Java Servlet technology, use XML, eXtensible HyperText Markup Language (XHTML) and Cascading Style Sheets (CSS). The PCMAT platform (Fig. 8.2) is based on a constructivist approach, and aims to assess the user knowledge and to present contents and activities adapted to the current needs and learning style of the student.

Several models have been used to implement AHS, such as the Dexter Model, Amsterdam Hypermedia Model, Adaptive Hypermedia Application Model (AHAM), or Munich Reference Model (Wu et al. 1999). The system architecture, presented in Fig. 8.3, is based on some strategies already used in the AHAM model.

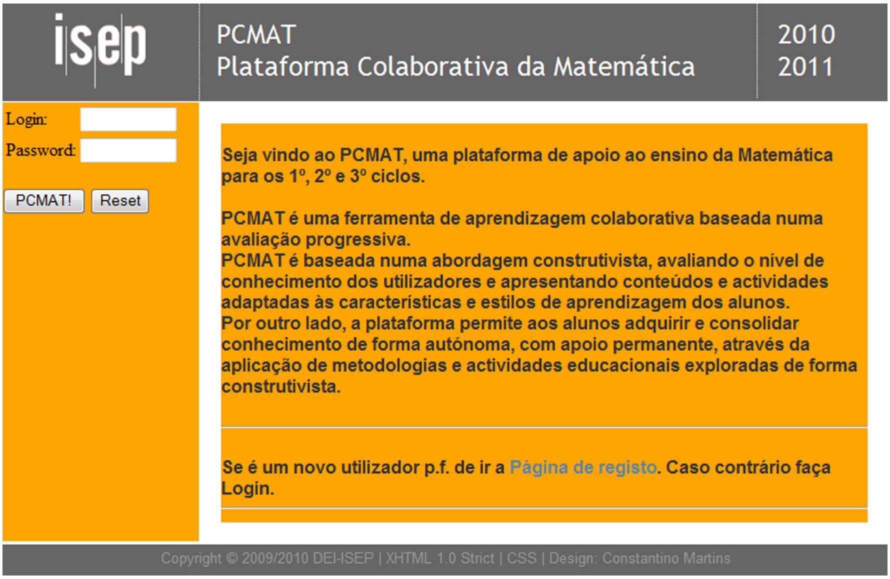


Fig. 8.2 PCMAT – Mathematics collaborative educational system

Therefore, in our system, the user requests an activity by clicking on a link in a Web page. Every page corresponds to a domain concept or a cluster of domain concepts. The system checks the suitability of the requested page for the current user. The adaptation rules used to check if the page is suitable are defined in the adaptation model. Updates to the SM are inferred from the interaction between the user and the application. The answers of the user allow the system to estimate the users knowledge level about the concepts related with the requested content.

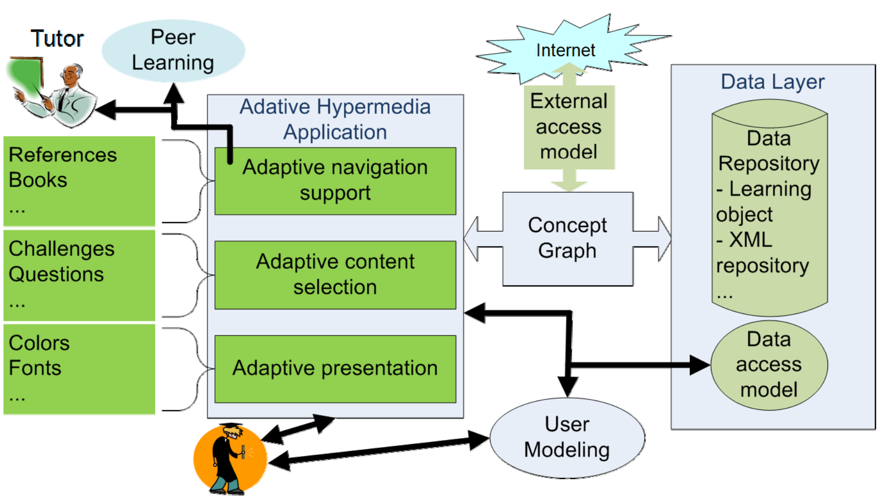


Fig. 8.3 System architecture

8.6.2 Student Model Implementation

Two different types of techniques can be used to implement the SM: Knowledge and Behavioral based (Kobsa 1993, Martins et al. 2008b). The Knowledge-based adaptation results for data collected through questionnaires and studies of the user, with the purpose to produce a set of initial heuristics. The behavioral adaptation results from monitoring of the user during his activity (Martins et al. 2008).

The use of stereotypes allows classifying users in groups and generalizes student characteristics to that group (Martins et al. 2008). The definition of the necessary characteristics for the classification in stereotypes must take in consideration the desirable granularity degree (Martins et al. 2008).

The approach is used to build the UM is the Stereotype Model with the overlay model for the knowledge representation of the student. The representation of the stereotype is hierarchical. Stereotypes for users with different knowledge have been used to adapt information, interface, scenario, goals, and plans.

The user modeling process starts with the identification of the user subgroup using questionnaires and learning styles (Fig. 8.4), then the identification of key characteristics (each one to identify the members of a user-subgroup), and finally the representation in hierarchical ordered stereotypes with inheritance. We also use the reliability analysis in Software for Statistical Analysis (SPSS) to compute the Cronbach's alpha reliability coefficient in every questionnaire (Woodward et al. 1983). Related to the learning style questionnaire, the value of the Cronbach's alpha coefficient is 0.91. So we can affirm that the internal consistency is very good, suggesting that the items have high internal consistency.

The user plan is a sequence of user actions that allows him to achieve a certain goal. The System observes the user actions and tries to infer all possible user plans. This goal is possible because our system has a library of all possible user actions and the preconditions of those actions.

A large number of criteria are established in the Stereotype definition depending on the adaptation goals (Martins et al. 2008a, Martins et al. 2008b). The definition of the characteristics of the student account the Domain Model and the constructivist approach of the application. For example, Table 8.3 presents a generic student profile used by PCMAT. The tools used to collect data are (Fig. 8.5):

- For the DID:
  - Questionnaires,
  - Certificates,
  - Curriculum Vitae,
  - Learning Styles,
  - Psychological exams.
- For the DDD:
  - Questionnaires,
  - Exams.

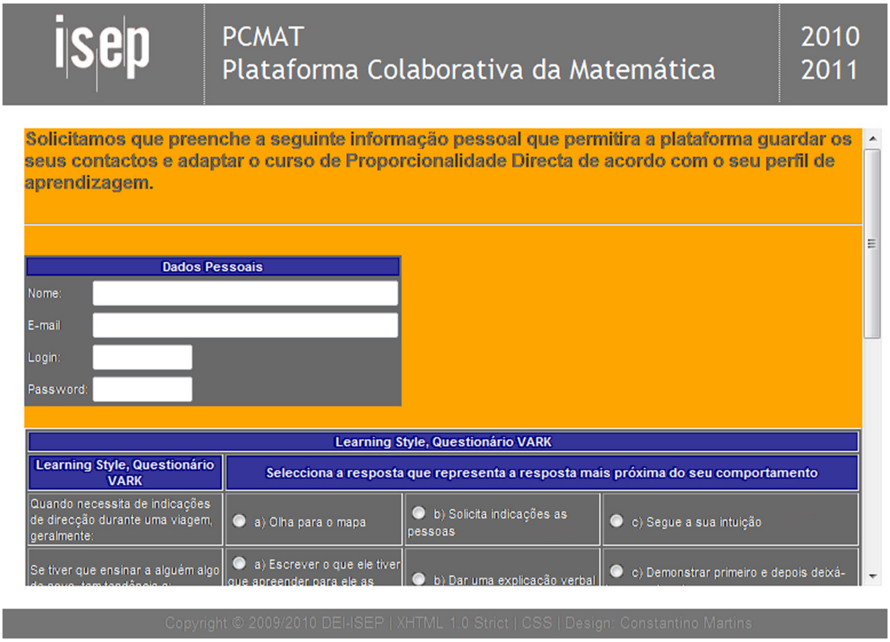


Fig. 8.4 Questionnaires and learning preferences

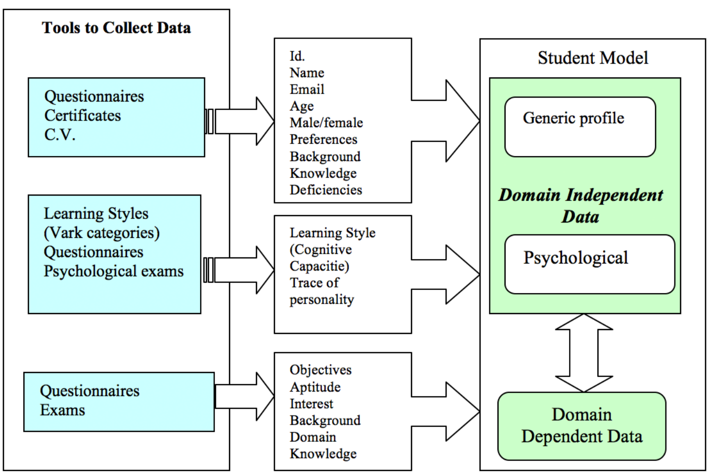


Fig. 8.5 UM architecture

For the definition of the Learning Styles of the student we are using the Kolb Learning Styles Matrix (Table 8.2). Concerning that and the objective of Domain Dependent Data, users aptitude and assessments result will be monitoring (Fig. 8.6). For each student profile, PCMAT keep an XML file.

Table 8.3 Characteristics used in the SM

Model	Profile	Characteristics
Domain Independent Data	Generic Profile	Personal information
		Demographic data
		Academics background
		Qualifications
		Knowledge (background knowledge)
		Deficiencies: visual or others
		Application domain
	Psychological profile	Learning style
		Cognitive capacities
		Traces of the personality
		Inheritance of characteristics
Domain Dependent Data		Objectives
		Planning / Plan
		Complete description of the navigation
		Knowledge acquired
		Results of evaluations
		Context model
		Aptitude
		Interests
		Deadline extend

This XML file contain the data related to the DDD and DID. The structure and type of the data are validated by the SM PCMAT Schema (see Sect. 8.1).

Every activity corresponds to a set of concepts in the Domain Model and in the UM (implemented through an overlay model). Each concept has an attribute to represent the student's knowledge. The value of the knowledge attribute is an integer between 0 and 100. The knowledge value about each concept is updated by the interaction model (Sect. 8.6.4 - domain and interaction model). Also, the learning style for each student is updated by the interaction model as well.

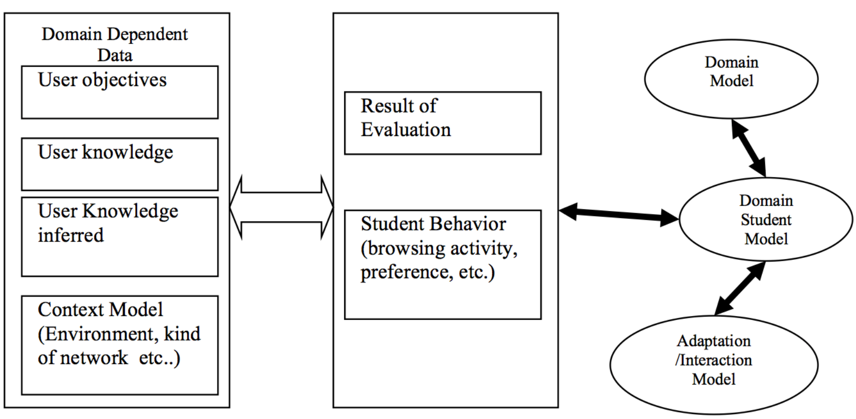


Fig. 8.6 Domain dependent data architecture of our UM

The requested page presentation is adapted by adaptation rules in two ways:

- Information content of the page can be changed (e.g., by conditionally including or hiding fragments),
- Links in the page can be manipulated. Links to pages that are considered is not suitable and can be annotated (for example with a red marker) or can be hidden; in addiction, the link route can be changed as well.

System adaptation (adaptation to content or links) to the user can cause UM updates as well, as the code of SM schema outlined as follows:

```
<xsd:element name=" Student_Model ">
    <xsd:complexType>
    <xsd:sequence minOccurs="1" maxOccurs="1">
    <!--definition of data related with DDD and DID -->
    <xsd:element name="Domain_Independent_Data"
          type="TDomain_Independent_Data"/>
     <xsd:element name="Domain_Dependent_Data"
          type="TDomain_Dependent_Data" />
    </xsd:sequence>
    </xsd:complexType>
</xsd:element>
<xsd:complexType name=" TDomain_Independent_Data ">
    <xsd:sequence>
    <xsd:element name="Generic_Profile"
          type="TGeneric_Profile" />
    <xsd:element name="Cognitive_Profile"
          type="TCognitive_Profile" />
    </ xsd:sequence>
</xsd:complexType>
<xsd:complexType name=" TGeneric     Profile ">
    <xsd:sequence minOccurs="1" maxOccurs="1">
    <xsd:element name="Personal_Information"
          type="TPersonal_Information" />
    <xsd:element name="Academic_Background" type="TAcademic" />
    <xsd:element name="Demographic_data"
          type="TDemographic_data" />
    <xsd:element name="Background_Knowledge"
          type="TBackground_Knowledge" />
    </xsd:sequence>
</ xsd:complexType>
<xsd:complexType name=" TDomain_Dependent_Data ">
    <xsd:sequence>
    <xsd:element name="Domain_Knowledge"
          type="TBackground_Knowledge" />
    <xsd:element name="Task made" type="TTask_made" />
    ...
...
    </xsd:sequence>
 </xsd:complexType>
 ...
```

8.6.3 *Authoring Tool for Metadata for the Learning Objects*

With the aim of teachers and developers to create metadata instances pertaining to the respective LO it was built a web application. There are tools available freeware, as Reload (http://www.reload.ac.uk/) or LomPad (http://helios.licef.ca:8080/LomPad/en/ in- dex.htm), which allow the writing of metadata instances compliant to IEEE LOM. However, they are desktop applications and do not hold a Portuguese version. Hence, we acknowledge the need to develop our own metadata authoring tool, as the one pictured in Fig. 8.7.

This application is developed in Java and uses an XML document as its configuration file. This XML file contains an instance of every element identified in the IEEE LOM and it may contain other elements alien to the standard. For instance, the element used to point out the compatibility of a learning object with a learning style is one of these alien elements. To each element, several attributes were added that determine several configurations options undertaken by the application and, one in particular, determines if a certain element will be used or not to characterize a learning object. Whenever feasible some elements are present pre-filled with default values and others have a vocabulary list from where to choose one or more values, even if in some cases such is not foreseen in IEEE LOM.

Keyword is an example of such an element. In order to provide the users of the platform with the least noise possible during the process of searching the PCMAT LO database, the keywords associated with each learning object are selected from a list build from a mathematical thesaurus. Fig. 8.8 shows a screen of the popup window containing the keywords list.

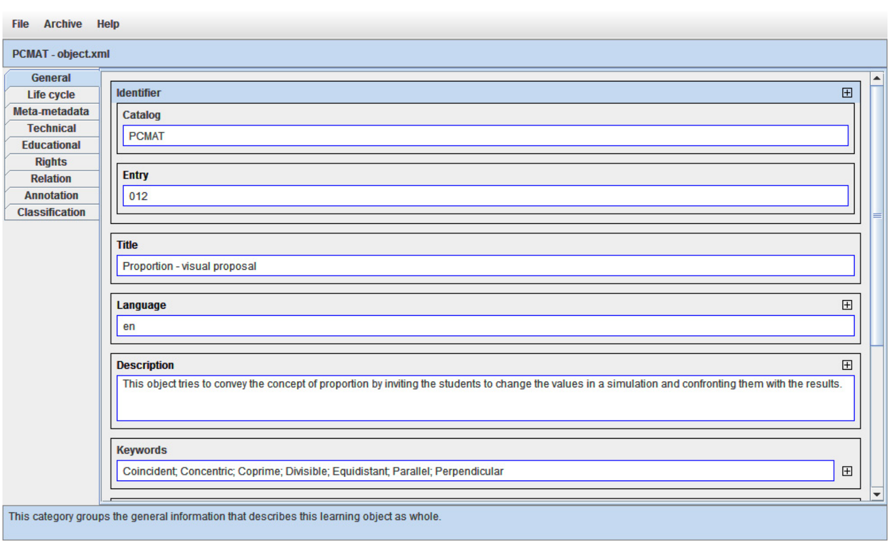


Fig. 8.7 PCMAT metadata application

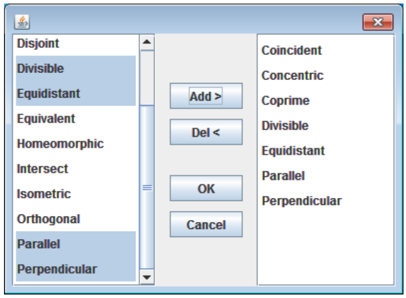


Fig. 8.8 PCMAT metadata application – keywords list

8.6.4 Domain and Interaction Models Development

The Domain Model represents concept hierarchies and the related structure used to represent an estimation of the user knowledge level, by means of a quantitative value. The Domain and Adaptation Models use the student characteristics represented in the SM. The knowledge about the user, represented in the SM, is used by the Adaptation Model to define a specific domain concept graph, adapted from the Domain Model, in order to address the current user needs.

The path used in the graph is defined by: the interaction with the student using a progressive assessment, the student knowledge representation defined by the Overlay Model and the user characteristics in the SM.

The system adaptation (adaptation of contents or links) to the user can produce UM updates as well. The results of Domain and Adaptation Models achieved are: The development of the concept graph by each user to use in the Adaptation rules and the Definition of the Adaptation Model using the characteristics of the student in the UM.

The Interaction Model represents and defines the interaction between the user and the application (Martins et al. 2008a). The Interaction Model enables the system to present the following functionalities, which are sketched in Fig. 8.9:

- To change the content presentation,
- To manipulate the structure of links or the links annotation with the objective to allow the students to reach the learning goals proposed in their training,
- To guide the user to the relevant information and keep him away from the irrelevant information or pages that he still would not be able to understand, it is used the technique generally known by link adaptation (hiding, disabling, and removal).

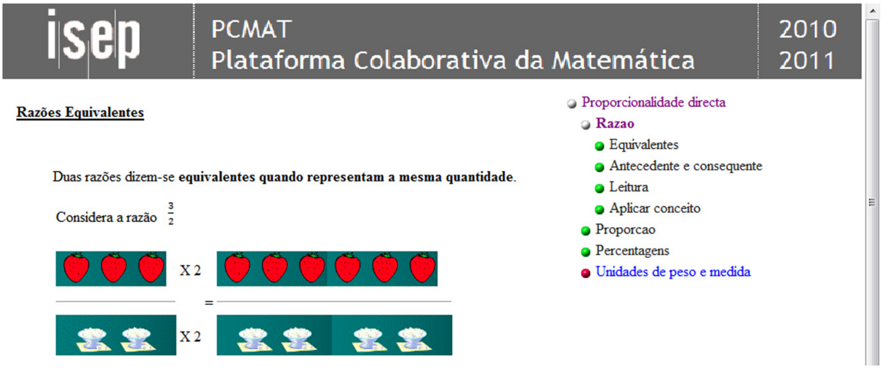


Fig. 8.9 Example of a PCMAT content

The platform in the content page supplies additional or alternative information to certify that the most relevant information is shown. The technique that is used for this task is generally known by content adaptation. The Domain Model represents concept graph hierarchies. The concept graph is set in a XML file as follows:

```
<concept_hierarchies>
      <concept_relation>
            <concept_name>A</ concept_name>
            <hierarchy>
                  <firstchild>A1</ firstchild>
                  <nextsibling>A2</ nextsibling>
                  <!-- root element of the concept graph !-->
                  <parent>proporcionalidade</ parent>
            </hierarchy>
            <children>
                  <concept_name>A1</ concept_name>
                  <concept_name>A2</ concept_name>
                  <concept_name>A3</ concept_name>
                  <concept_name>A4</ concept_name>
            </children>
            ...
      </concept_relation>
      <concept_relation>
            <concept_name>A1</ concept_name>
            <hierarchy>
                  <firstchild></ firstchild>
                  <nextsibling> A2</ nextsibling>
                  <parent>A</ parent>
            </hierarchy>
            <children></ children>
      </concept_relation>
</ concept_hierarchies>
```

The knowledge about the user, represented in the SM, is used by the Adaptation Rules Model to define a specific domain concept graph, adapted from the Domain Model, in order to address the current user needs.

The path used in the graph is defined by:

- The interaction with the student using a progressive assessment,
- The student knowledge representation defined by the Overlay Model,
- The user characteristics in the SM.

The Adaptation Rules Model is defined in a XML file according to a couple of items: 1) for each concept PCMAT have a set of attributes; 2) each attribute is related with a set of adaptation rules. As regards with the attribute concept, it contains: a default value and a list of rules (i.e., set of adaptation rules). A rule holds: a Boolean expression to involve attributes of concepts or attributes of the SM (i.e., this expression must be true for the resulting sequencing action to be triggered) and a rule action that can update some attributes of concepts or attributes of the SM.

When the user tries to access a resource page (concept) the next events are triggered:

- The access attribute of the concept is accessed,
- All the rules of access attribute are evaluated and activated,
- One of these rules must evaluate the value of the suitability attribute of the concept,
- Another of these rules is responsible to assign values to the resource visibility attribute of the concept.

The suitability attribute is used to define if a page (concept) is suitable to the learner. The value of this attribute results from the evaluation of a condition expressing the prerequisites to access the current concept. These prerequisites are defined in the pedagogical model and are formed by the minimum knowledge levels the student attained in a set of concepts. This pedagogical model is defined by teachers and is also implemented through a XML file. The following rule shows the mechanism used to update the suitability attribute:

```
<rule>
      <condition>
            <!-- Condition definition
      </condition>
      <rule_effect>
            <concept>
                  <name>concept_name</name>
                  <attribute> suitability </attribute>
                  <value>true</value>
            </concept>
      </rule_effect>
</rule>
```

If the value of the attribute *concept.suitability* associated to a page content is false then the page is not shown. Being the case, the student is conducted to other contents or to specific content fragments. The attribute knowledge receives a value between 0 and 100, and is used to represent an estimative of the knowledge level about a particular concept.

This attribute is updated during student activities, and can be used, for instance, to make a decision about to show or hide fragments, or to adapt links (hiding links or adding annotations to links).

We rely on the constructivist approach to suggest contents and activities to the student according with his behavior in previous activities (Martins et al. 2008a). Each activity or content (page, content fragment, etc) is associated to a prominent learning preference. The learning preference associated to the student is also represented in the SM through the attributes *personal.lst* (learn by reading and hearing), *personal.lsv* (visual learning) and *personal.lsp* (practical learning). When the student uses the system for the first time, these attributes are initialized from the results of the learning styles questionnaire (Sect. 8.6.2).

When the student accomplishes successfully an activity in PCMAT, the knowledge level of the concepts involved is updated by the following mechanism:

```
Let A1,A2,A3,...,An be the set of concepts associated with the activity
For each i in {1,2,3,...,n}
      Ai.knowledge = min(Ai.knowledge+Ai.knowledge*0.25,100)
Let B1,B2,B3,...,Bm be the set of concepts from which concept Ai depends
For each i in {1,2,3,...,m}
      Bi.knowledge = min(Bi.knowledge+Bi.knowledge*0.1,100)
```

A similar mechanism is used to update the learning preference of the student. The following example shows how the learning preference attributes are updated when the prominent learning preference attribute associated to the activity is *personal.lst*:

```
personal.lst = min(personal.lst + 1,10)

If personal.lsv >= personal.lsp then personal.lsv= max(personal.lsv-1,0)

If personal.lsv < personal.lsp then personal.lsp = max(personal.lsp-1,0)
```

The same algorithm is used in case of an activity whose learning preference is *personal.lsv* or *personal.lsp*. In case of student failure in an activity, a similar approach is used to downgrade the concept knowledge level and the learning preference attribute. The PCMAT has an authoring module to make questions and automatic generation of tests. The front-end of the application is developed using XHTML, CSS and Javascript, and the back-end is developed in Java.

This feature allows users to make test questions which are simple or parameterized. All questions are related to at least one concept and at most to five concepts. Each question is classified according its compatibility to a learning preference.

The body of the question is directly inserted by the user (Fig. 8.10), but it is also introduced by uploading a text file with the question. A resource, such as an image file, may be added as well. The user decides whether the question will be a multiple choice question or a restricted answer question (Fig. 8.11). In the case of restricted answer questions, the correct answer can be given in several ways, that is to say the word order may differ. It is therefore necessary to parse the answer, using a probabilistic natural language parser, to verify if it is correct or not.

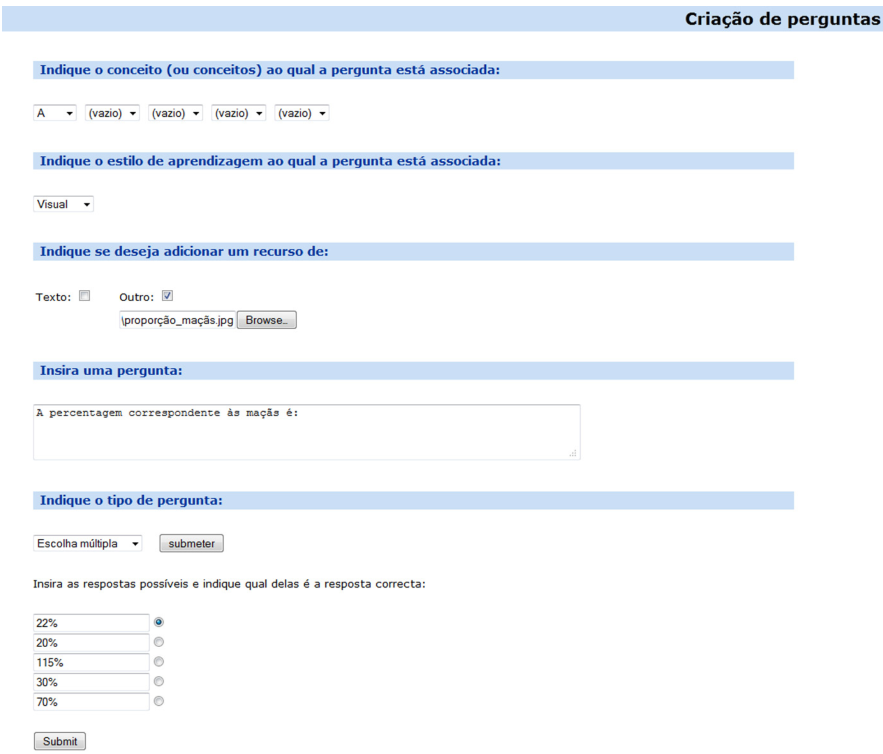


Fig. 8.10 User interface for the creation of questions

The creation of parameterized questions (Fig. 8.12) requires the user to comply with certain rules. For instance, he/she can only utilize a maximum of five different variables which must have specific names and structure. This makes it possible to find the variables in the text and replace them by the given parameters. A table is provided for the insertion of parameter values, thus ensuring each parameter will be correctly associated with each variable.

Aside from adding new questions to the system, the user can also use it to generate tests. He/she must input the number of questions the test should be composed of and the concepts to which those questions are related. The system will create a test by randomly choosing and retrieving questions from the database (Fig. 8.13).

If a chosen question happens to be a parameterized question, the system will randomly choose one of the possible sets of parameters and replace the questions variables with their respective values. After all the questions have been chosen the system creates an XHTML file with the finalized test.

The results achieved with the PCMAT Domain and Adaptation Model are two: the development of the concept graph by each user to use in the adaptation rules and the Definition of the Adaptation Model using the characteristics of the student in the UM.

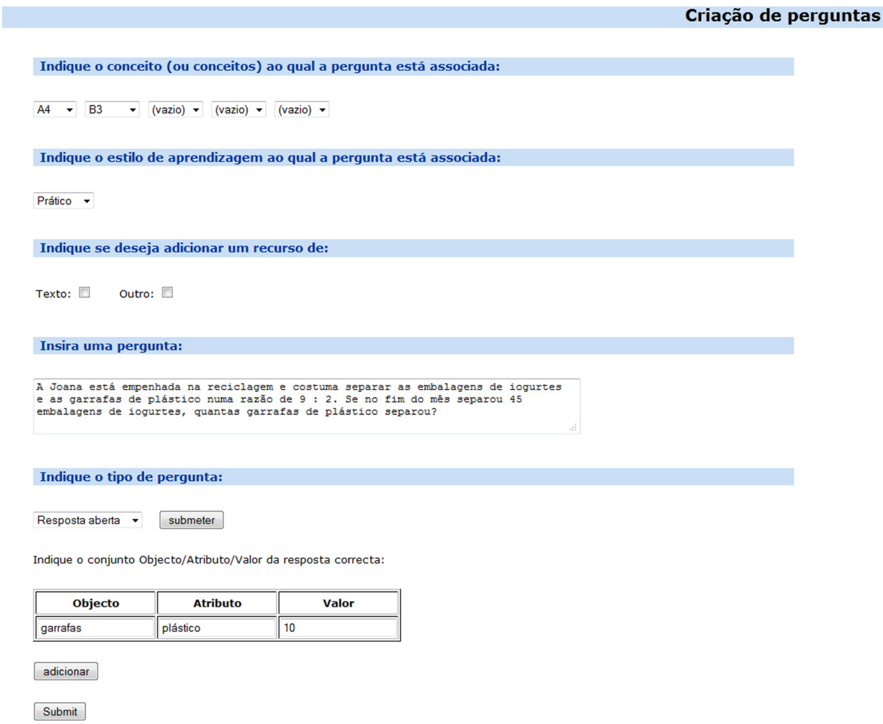


Fig. 8.11 Creation of a restricted answer question

Resuming, the features of PCMAT interaction model are: 1) show different content in different formats; 2) to define the structure of the links or the links annotation; 3) to guide the student to the relevant information and keep him away from the irrelevant information or pages; 4) to supply, in the content (page), additional or alternative information; 5) activities in different formats (Word, PDF, Flash and others), using LO; 6) assessments adjusted by difficulty, knowledge and profile of the student; 7) news about national mathematics events; 8) online references.

8.7 Results Analysis

The learning domain chosen by the teachers to perform the first evaluation of the system was Direct Proportion unit, included in the mathematics course in the 7th grade. The students are 12 and 13 years old. The first version of the framework was already implemented, tested and evaluated in learning processes in two mathematics basic schools: 1) one class with 26 students of the 7th basic year from the first school, 2) two classes, one with 24 students and the other with 27 students of the 7th grade from the second school.

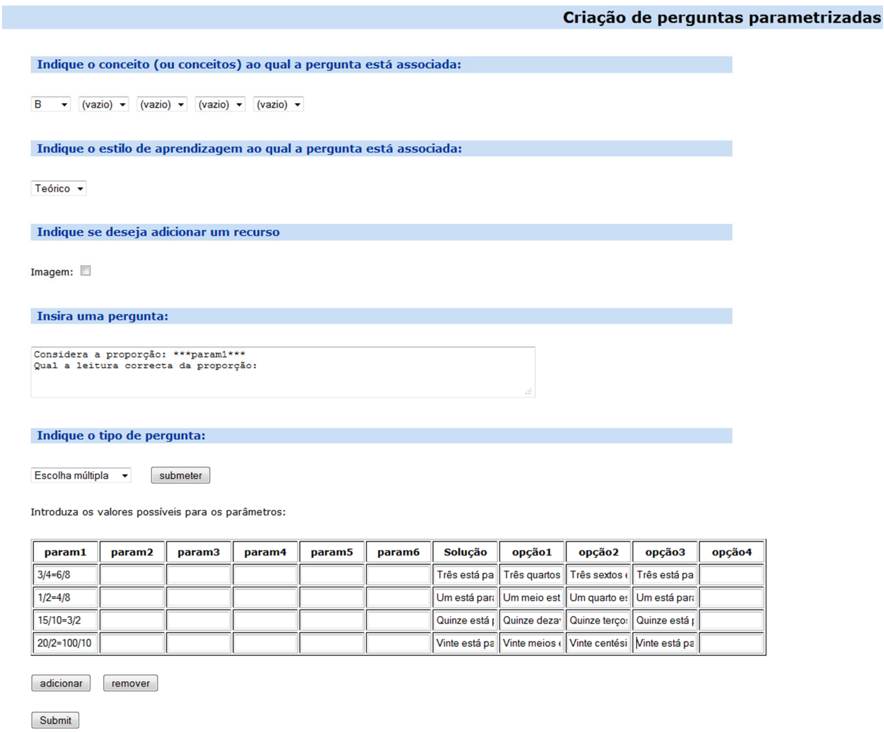


Fig. 8.12 Creation of a parameterized question

None of the students have previous experience of using some kind of AHS, but more than 75% of them were familiar with personal computers (PC). Generally, they use PC for the Internet browsing and for playing the games.

The first step of the evaluation was to divide randomly each class into two groups: experimental and control group. The random process has some criteria to select and put the students in each group:

- If possible each group must have the same students number (or approximate),
- If possible each group must have the same number (or approximate) of excellent students. For this effect, the students have taken a diagnostics test,
- Each group have a similar number of students with the same learning preference; for this effect a learning style questionnaire was made (see Sect. 8.6.2),
- Each group has a similar distribution of student gender.

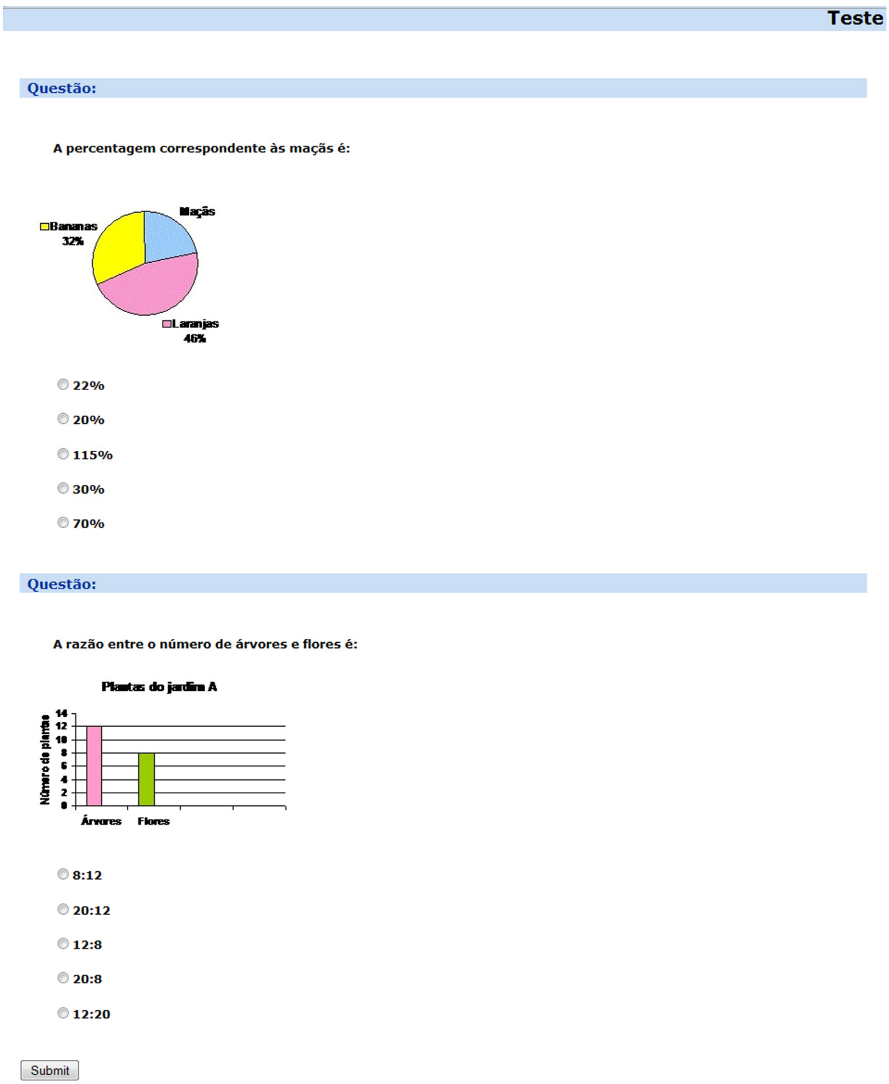


Fig. 8.13 Example of a XHTML test composed of one question

The results of this first step were the following:

- Creation of two groups in the first school (experimental and control group); each group has 13 students. Also each group has one teacher to follow the learning process,
- Creation of two groups in the second school (experimental and control group). The group using PCMAT has 26 students. Also each group has one teacher to follow the learning process,
- Analyzing the data of each group related to the learning preference, 58% of students of the experimental group have a visual learning preference. In the control group the value of visual learning preference was 62%. Some case studies have demonstrated that: learning style in Basic Schools is more visual.

Thus, 39 students of the experimental group used PCMAT to learn Direct Proportion during fifteen days; and 37 students (control group) learnt Direct Proportion with the traditionally learning methodology during fifteen days. Using 4 different teachers for reducing the correlation between the quality of the learning process and the competence of the teacher.

The second step was to use questionnaires to collect data for the PCMAT SM (DDD and DID, but some data have already been collected in the first evaluation step). The second step was applied to all the students with the goal to validate the data. The third evaluation step was to make a final test to all students. The test is the same for the entire students (experimental and control group).

The collected evaluation data showed the next findings:

- In the first school the average of student scores in the experimental group is higher than the average of student scores in the control group, mean $\mu = 62.0$ ($\sigma = 17.0$ (standard deviation)) vs. $\mu = 55.3$, but the observed differences are not statistically significant ($p = 0.281$). The two groups were statistically compared using a two sided, independent samples t test with a 0.05 (5%) critical level of significance ($t = 1.10$, degrees of freedom = 24),
- In the second school, the average of student scores in the experimental group is higher than the average of student scores in the control group; $\mu = 61.5$ ($\sigma = 21.5$) vs. $\mu = 54.0$ ($\sigma = 14.9$), but the observed differences are not statistically significant ($p = 0.157$). The two groups were statistically compared using a two sided, independent samples t test with a 0.05 (5%) critical level of significance ($t = 1.44$, degrees of freedom = 48),
- And, analyzing simultaneously both schools, the average of student scores in the experimental group is higher than the average of student scores in the control group, $\mu = 61.7$ ($\sigma = 19.9$) vs. $\mu = 54.4$ ($\sigma = 14.3$). Although the observed differences are not statistically significant ($p = 0.073$) the difference between groups seems to be clear. Again, the two groups were statistically compared using a two sided, independent samples t test with a 0.05 (5%) critical level of significance ($t = 1.82$, degrees of freedom = 74).

These values are good indicators and may allow us to conclude about the adequate features and effectiveness of PCMAT system. Nevertheless, additional results, with an increased sample size, will allow validating these assumptions.

Students also perceived this tool as very relevant for their learning, as a self-operating application to be integrated in a more global learning strategy that includes also tutoring (direct contact with the teacher) and peer learning. Teachers agreed with these definitions of the platform, as well.

Also, the collected evaluation data showed that the development of two authoring tools, the authoring tool for metadata of the LO (Sect. 8.6.3) and the authoring tool for the creation of questions and automatic generation of tests (Sect. 8.6.4), showed a very high degree of interest and motivation from teachers alike, resulting from its use.

Another result was the definition of a new strategy and architecture for the implementation of an Educational AHS in basic schools in Portugal (see Sect. 8.6.1). The capacity of adaptation of these tools in relation to the different needs and the diversity of the background of each student is necessary for bigger effectiveness and efficiency of the learning process.

The main result of the present development is the definition and validation of the characteristics of the student to be stored and the selection of the techniques of the Overlay Model and stereotypes for the representation of the user knowledge in the SM (see Sect. 8.6.2). Moreover, the present work allowed defining an adaptation and interaction model (see Sect. 8.6.4). Next school year, PCMAT will be tested in another school and also in the two previous schools.

8.8 Conclusions

In the scientific area of UM, numerous research and developed systems already seem to promise good results (Kules 2000), but yet some experimentation and implementation are still necessary to conclude about the utility of the UM. That is, the experimentation and implementation of these systems are still very scarce to determine the utility of some of the referred applications.

The intelligent behavior of the learning platform is based on the existence of a tentative description of the student – the SM. The contents of this model and the student most prominent learning style are used by an interaction model to select the most appropriate response to student actions.

The difficulty in building a SM using an Overlay Model for a given student depends on the type of information we want to store in the model. The main result of the present development is the validation of a SM that will allow the support of adaptive functionalities based on the use of LOM standard to truly support a constructivist learning and cognitive path. The number and type of characteristics to use in the SM depend on the finality of each system, but some relevance is in the cognitive part, learning styles and student knowledge.

The analysis, application, implementation, integration and evaluation of techniques used to adapt the presentation and navigation in educational AHS, using metadata for the LO and user modeling, will contribute to improve the value and implementation of e-learning in Basic Schools, in a way to make possible the educational process more adaptive to the student learning preference.

The capacity of the adaptation of these tools, considering the different necessities and the diversity of individual information source of each student will be necessary, namely for more and more efficiency in the learning process.

It will also be possible to introduce more responsibility to the student in his learning process, namely in the individualization and adaptability of learning.

The PCMAT project allowed us to define new strategies for the implementation of an AHS to support and improve Mathematics in basic Schools context. Additional contributes of the project includes the definition of a SM describing the information, knowledge, preferences, and learning style of the user; the definition of a process and the tools needed to produce LO aligned with the LOM standard; and the implementation of a set of adaptive and dynamic pedagogical strategies.

The outcome results are good indicators and may allow us to conclude about the adequate features and effectiveness of an AHS to improve e-learning based systems in a basic school environment in mathematics. However, further experiments will be necessary to confirm these results.

One of the next steps in the development of the platform will be the inclusion of a chat environment that will allow capturing the messages changed between learners during the learning process. The goal is to infer learners' doubts from the messages. For that we will use mechanisms able to explore natural languages sentences and clustering algorithms to identify difficulties among the students. This feature will turn the platform into a full collaborative platform, where students may share their difficulties and to get appropriate feedback.

Acknowledgments. The authors would like to acknowledge FCT, FEDER, POCTI, POSI, POCI, and POSC for their support to GECAD unit, and the project PCMAT (PTDS/CED/108339/2008).

References

Barker, P., Campbell, L.M.: Metadata for learning materials: an overview of existing standards and current developments. Journal of Technology, Instruction, Cognition and Learning 7(3-4), 225–243 (2010)

Benyon, D.: Adaptive systems: A solution to usability problems. Journal of User Modeling and User-Adapted Interaction 3(1), 65–87 (1993)

Brusilovsky, P.: Adaptive Hypermedia: An Attempt to Analyze and Generalize. In: Brusilovsky, P., Kommers, P., Yuan, F. (eds.) MHVR 1994. LNCS, vol. 1077, pp. 288–304. Springer, Heidelberg (1996)

Brusilovsky, P.: Adaptive hypermedia. Journal of User Modeling and User-Adapted Interaction 11(1-2), 87–110 (2001)

Chepegin, V., Aroyo, L., De Bra, P., Heckmann, D.: User modeling for modular adaptive hypermedia. In: Proceedings of Applications of Semantic Web Technologies for Educational Adaptive Hypermedia Workshop (2004)

Currier, S.: Metadata for learning resources: An update on standards activity for 2008. Ariadne (2008), `http://www.ariadne.ac.uk/issue55/currier` (accessed December 8, 2011)

DCMI. Dublin core education application profile task group. Resource document. Dublin Core Metadata Initiative Limited (2007), http://dublincore.org/educationwiki/DC-Education%20Application%20Profile%20Task%20Group (accessed October 7, 2011)

DCMI. Guidelines for Dublin core application profiles. Resource document. Dublin Core Metadata Initiative Limited (2008), http://dublincore.org/documents/profile-guidelines/ (accessed December 7, 2011)

DCMI. Dublin core metadata element set, version 1.1. Resource document. Dublin Core Metadata Initiative Limited (2010), http://dublincore.org/documents/dces/ (accessed December 8, 2011)

De Bra, P., Aroyo, L., Chepegin, V.: The next big thing: Adaptive Web-based systems. Journal of Digital Information 5(1) (2004)

Fleming, N.: Teaching and learning styles VARK strategies. Neil Fleming, Christchurch (2001)

IEEE. IEEE standard for learning object metadata. Resource document. Learning Technology Standards Committee of the IEEE Computer Society (2002), http://ltsc.ieee.org/wg12/files/LOM_1484_12_1_v1_Final_Draft.pdf (accessed December 2, 2011)

IEEE. IEEE standard for learning technology-extensible markup language (xml) schema definition language binding for learning object metadata. Resource document. Learning Technology Standards Committee of the IEEE Computer Society (2005), http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1532505 (accessed December 2, 2011)

Jonassen, D.: Objectivism versus constructivism: Do we need a new philosophical paradigm? Journal of Educational Technology Research and Development (1991), doi:10.1007/BF02296434

Kavcic, A.: The role of user models in adaptive hypermedia systems. In: Proceedings of MELECON 2000, pp. 119–122 (2000)

Kobsa, A.: User modeling: Recent work, prospects and hazards. In: Schneider-Hufschmidt, M., Kühme, T., Malinowski, U. (eds.) Adaptive User Interfaces: Principles and Practice, pp. 111–128. North Holland Elsevier, Amsterdam (1993)

Kolb, A.Y.: Learning styles and learning spaces: Enhancing experiential learning in higher education. Journal of Academy of Management Learning & Education 4(2), 193–212 (2005)

Kules, B.: User modeling for adaptive and adaptable software systems. Resource document. Learning Technologies (2000), http://otal.umd.edu/UUGuide/wmk/ (accessed December 2, 2011)

Martins, C., Faria, L., Carvalho, C.V., Carrapatoso, E.: User modeling in adaptive hypermedia educational systems. Journal of Educational Technology & Society 11(1), 194–207 (2008)

Neven, F., Duval, E.: Reusable learning objects: a survey of LOM-based repositories. In: Proceedings of MULTIMEDIA, pp. 291–294. ACM, New York (2002)

OAI. Open archives initiative: The open archives initiative protocol for metadata harvesting, version 2.0. Resource document. Open Archives Initiative (2008), http://www.openarchives.org/OAI/2.0/openarchivesprotocol.htm (accessed October 21, 2011)

Reichmann, S.W., Grasha, A.F.: A rational approach to developing and assessing the construct validity of a student learning style scale instrument. Journal of Psychology 87, 213–223 (1974)

Ritu, D., Sugata, M.: Learning styles and perceptions of self. Journal of International Education 1(1), 61–71 (1999)

Schmeck, R.R.: Learning styles of college students. In: Dillon, R., Schmeck, R. (eds.) Individual Differences in Cognition, pp. 233–279. Academic Press, New York (1983)

Stash, N., Cristea, A., De Bra, P.: Explicit intelligence in adaptive hypermedia: Generic adaptation languages for learning preferences and styles. In: Proceedings of Workshop CIAH of HT 2005, pp. 75–84 (2005)

Vassileva, J.: A task-centered approach for user modeling in a hypermedia office documentation system. Journal of User Modeling and User-Adapted Interaction 6(1), 185–223 (1996)

Woodward, C.A., Chambers, L.: Guide to questionnaire construction and question writing. Canadian Public Health Association, Ottawa (1983)

Wu, H., Houben, G.J., De Bra, P.: User modeling in adaptive hypermedia applications. In: Proceedings of INFWE 1999, pp. 10–21 (1999)

Abbreviations

AHA	Adaptive Hypermedia Architecture System
AHAM	Adaptive Hypermedia Application Model
AHS	Adaptive Hypermedia System
CSS	Cascading Style Sheets
DCAP	Dublin Core Application Profile
DCMES	Dublin Core Metadata Element Set
DCMI	Dublin Core Metadata Initiative
DDD	Domain Dependent Data
DID	Domain Independent Data
LO	Learning Object
LOM	Learning Object Metadata
SM	Student Model
SPSS	Software for Statistical Analysis
UM	User Modeling
XHTML	eXtensible HyperText Markup Language
XML	eXtensible Markup Language

Chapter 9
A Framework for Automatic Construction of Reusable Adaptive Courses: The Case of ProPer SAT 2.0

Ioannis Kazanidis[1] and Maya Satratzemi[2]

[1] Kavala Institute of Technology
65404, Kavala, Greece
kazanidis@teikav.edu.gr

[2] University of Macedonia
54006, Thessalonniki, Greece
maya@uom.gr

Abstract. This chapter puts forward a proposal for a framework that enables the automatic construction of adaptive and reusable courses. The case study of the implementation of an adaptive Learning Management System (LMS) and an authoring tool, named ProPer and ProPer Sharable Content Object Reference Model (SCORM) Authoring Tool (SAT) 2.0 are presented. ProPer delivers SCORM compliant courses and ProPer SAT 2.0 helps authors to construct SCORM courses quickly that can be automatically adapted to user learning style. Additionally, ProPer SAT 2.0 incorporates intelligent functionalities since it has the ability to propose that appropriate content be inserted into a course, according to the domain, difficulty level and user ranking of the proposed content. Evaluation results have shown that end-users find it easy and useful and intend using it in the future.

9.1 Introduction

Not only have times changed but also situations and prospects. Information and Communication Technologies (ICT) have brought about new capabilities which provide new opportunities for all. People with different characteristics, time, and space restrictions are in the process of increasing their knowledge through distance and e-learning. Fortunately, the educational process has improved and is now able to provide personalized instruction through the use of Adaptive Educational Hypermedia Systems (AEHS). Since, however, the development of AEHS, is not based on any common framework, the educational material produced cannot be reused by other systems, nor is the course from one system compatible with another.

In order to overcome this problem, developers strive to incorporate standards, such as SCORM, for the reusability, interoperability and durability of the

A. Peña-Ayala (Ed.): Intelligent and Adaptive ELS, SIST 17, pp. 213–238.
springerlink.com

educational content. Nevertheless, it is a difficult task for teachers without programming knowledge to design and author adaptive courses.

We believe that a good solution to the above problem is to combine AEHS with SCORM. Thus, while learners can take advantage of the personalized instruction offered, authors will not need to spend so much of their time on course authoring. Furthermore, we propose to extend adaptivity of the system to user learning style in order to increase the learning outcome even further. In previous works (Kazanidis and Satratzemi 2008; Kazanidis and Satratzemi 2009a) we combined the adaptive features of AEHS with the adoption of the SCORM standard and its specifications. Furthermore, we promoted the concept that it is possible for native SCORM compliant courses to be adapted to user learning style (Kazanidis and Satratzemi 2009b). Still, the development of adaptive SCORM compliant courses requires knowledge of the SCORM framework, learning style theories, and programming languages such as JavaScript and Hypertext Markup Language (HTML). Unfortunately, there is a lack of easy authoring tools for non-programmers and these teachers come up against serious difficulties in the development of such courses.

We propose a theoretical framework to enable the construction of adaptive and reusable courses. In order to evaluate our model and framework, an adaptive LMS and an authoring tool named ProPer and ProPer SAT respectively, were built.

ProPer delivers adaptive SCORM compliant courses and ProPer SAT helps authors construct SCORM courses quickly which can be automatically adapted to user learning style. Additionally, ProPer SAT in its second version (2.0) incorporates intelligent functionalities, having the ability to propose that appropriate content be inserted into a course, according to the domain, difficulty level and user ranking of the proposed content.

The chapter is organized as follows: Sect. 9.2 presents the theoretical background for our framework. Sect. 9.3 reviews related work in the area of adaptive systems and authoring tools for adaptive courses. Sect. 9.4 puts forward an explanation of the proposed framework for the development of intelligent authoring tools and the construction of adaptive courses. Sect. 9.5 presents the prototypes developed as a case study and a summary of the evaluation results. Finally, Sect. 9.6 offers conclusions and outlines possible avenues for future work.

9.2 Theoretical Background

The proposed framework combines technologies and characteristics from three domains of web based research. It uses adaptive technologies, such as AEHS, incorporates adaptation to user learning styles, and proposes the conformance with widespread technological standards such as SCORM. This Sect. briefly presents AEHS, some of the most popular learning style models as well as technological standards such as SCORM.

9.2.1 Adaptive Educational Hypermedia Systems

AEHS re-create user knowledge and characteristics in a way that provides personalized learning experience. They consist of three main components: 1) The Domain Model (DM) represents the system's domain knowledge; 2) the User Model (UM) depicts the user's knowledge of the domain as well as his/her individual characteristics; 3) the Adaptation Module (AM) describes how the adaptation will be applied and which items will be adapted.

According to Bruzilovsky 2001, adaptation is implemented using two major technologies: The first, Adaptive Presentation (AP) provides a variation to the content level. There are three main AP technologies: Adaptive text presentation, Adaptive multimedia presentation, and Adaptation of modality. The second technology, Adaptive Navigation (AN), adjusts the links or the course's link structure in order to steer the user towards certain links and away from others. The main AN technologies are: direct guidance, link hiding, link disabling, link removal, link sorting, link annotation, link generation, and hypertext map adaptation.

9.2.2 Learning Styles

According to Honey and Mumford (Honey and Mumford 1992) learning style refers to a person's habits and patterns of behavior that determines the desired means of learning. Some of the learning style models used by several AEHS are: Kolb's (Kolb 1984) experimental learning model; Honey and Mumford (Honey and Mumford 1992) based on Kolb's model; the Felder and Silverman (Felder and Silverman 1988) model; the Witkin's Field Dependent/Field Independent model (Witkin et al. 1977); and Gardner's (Gardner 1993) theory of Multiple Intelligences (MI). The systems we present in the case study support the construction of courses that provide adaptation consistent with the Honey and Mumford categorization based on Kolb's model.

According to Kolb's experimental learning model, learning is a process of knowledge construction through four distinct stages of a cycle: 1) concrete experience (CE); 2) abstract conceptualization (AC); 3) reflective observation (RO); and 4) active experimentation (AE). The student starts from any point in the cycle and continue in that order to pass through all the stages. More specifically, two pairs of variables make up student learning style preference, presented as two axes, each, with 'conflicting' modes at either end. One axis pivots between CE and AC, and the second between AE and RO. This model distinguishes learners into four categories each representing the combination of two preferred styles: Divergers (CE-RO), Assimilators (AC-RO), Convergers (AC-AE) and Accommodators (CE-AE).

Similar to Kolb's learning cycle the Honey and Mumford model, consists of four stages: 1) having an experience; 2) reviewing the experience; 3) concluding. from the experience; and 4) planning the next step.

The student can start from any point in the cycle and move on to the others. Each stage is related to a user's particular learning style. Thus, the corresponding Learner styles are: activist, reflector, theorist, and pragmatist.

9.2.3 Technological Standards and Specifications

Technological standards and specifications are set to facilitate the description, packaging, sequencing and distribution of content, learning activities, and learner information. The description and development of structured educational material comes via the metadata. Some of the most well-known standards for the creation of content are: SCORM (ADL 2011), LOM (Learning Object Metadata), IMS (Instructional Management Systems), AICC (Aviation Industry Computer-based training Committee), etc. SCORM has been the most widespread standard, which is based on Sharable Content Objects (SCO). The exploitation of this technology allows the use of educational material in multiple LMS and facilitates the discovery and reusability of such material (Duval 2000, Krull et al. 2006).

The SCORM standard was developed by the Advanced Distributed Learning (ADL) initiative and its main aim is to offer RAID (Reusable, Accessible, Interoperable, Durable) courses. It is comprised of a collection of specifications and standards for the development, packaging and distribution of the learning content. It is based on SCO, which are the smallest logical entities that can be delivered by a compliant course and which communicate with the LMS. Every SCO consists of information about the creation, discovery and aggregation of the appropriate educational resource in their most basic form called *assets*. Assets are digital media, such as text, images, video or any other digital data that can be delivered through a web based system. They are the smallest content piece in a SCORM course and can be reused by SCO through appropriate metadata.

SCORM holds there items (ADL 2011): 1) Content Aggregation Model (CAM), depicts the parts used in a learning experience, how to package them, depict them for discovery, and define sequencing rules; 2) Run-Time Environment (RTE) defines the LMS requirements for managing and communicating with the content, RTE provides an Application Program Interface (API) for communication between SCO and the LMS; 3) Simple Sequencing and Navigation (SSN) sets data and behaviors that a LMS uses to provide a learning experience.

9.3 Related Work

We present work relevant to our framework and the case study systems, which concerns to the outcome of adaptive courses. There are several frameworks on course construction and design. Tseng et al. 2008 set a modular framework to segment and transform content into modular learning objects based on SCORM.

In line with this framework, the contents of a course can be composed dynamically, according to the profile and portfolio of the individual students. Dagger et al. (Dagger et al. 2004) propose a framework for the development of adaptive courses which expands existing technological standards but this makes the courses not fully compliant with them. In addition, it does not provide reusable courses. Another framework proposed by Bradley (Bradley 2011) for learning module design does not support course adaptivity and reusability. Moreover Peng (Peng 2007) proposes a framework for enhanced adaptivity into SCORM courses.

However, these courses provide only sequence adaptation and not adaptive presentation of the educational content.

The framework proposed in this chapter supports the construction and delivery of adaptive courses and promotes easy authoring, and as a case study presents one adaptive LMS (ProPer), and one intelligent authoring tool (ProPer SAT). Currently, many systems incorporate adaptation to user learning styles, some of which, along with the learning model they implement, are featured below:

- INSPIRE (Papanikolaou et al. 2002) implements the Honey and Mumford model,
- TANGOW (Carro et al. 1999), LSAS (Bajraktarevic et al. 2003), CS383 (Carver et al. 1996) incorporate several dimensions of the Felder-Silverman model,
- AES-CS (Triantafillou et al. 2003) adopts the FD/FI model,
- EDUCE (Kelly and Tangney 2006) is based on Gardner's MI theory.

There are additional tools, which assist authors in the creation of adaptive courses. These systems allow the instructor to define specific rules in order to apply adaptation to the course or they provide ready instructional strategies. Following, we briefly present some such systems.

AHA! (Romero et al. 2005; Bra et al. 2006) is an open source web server extension that adds adaptation to applications, such as on-line courses. WELSA (Popescu 2008) is a system consisting of an authoring tool and a course player. By integrating characteristics from several models, it creates a unified learning style model (ULSM) taking into account some behavioral patterns for learner tracking functionality. MOT (Stash et al. 2004) is an online environment for the authoring of adaptive educational hypermedia. In MOT authors can either select an adaptive strategy that corresponds to an instructional strategy created by a different author and apply it to an arbitrary concept map or lesson map or define their instructional strategy. In its second version, called MOT 2.0 (Ghali et al. 2004), it focuses on collaborative authoring and social annotation between communities of authors.

Most of the above systems are inappropriate for teachers who are not technical and who have little or no programming knowledge. VIDET and REDEEM, however, come to their aid. VIDET (Armani 2005) is a visual authoring tool for designing adaptive courses, whose goal is to support easy course authoring for non-technical instructors and attempts to give the teacher full control over the adaptive operations being performed. It provides authoring tools for manipulating the hypertext structure, the content, the user model, and the adaptive interaction model.

REDEEM (Ainsworth et al. 2003) allows instructors with little technological background to import pre-existing courses and provides them with tools to define how they want to teach the material. Its focus is not on the construction of domain material but on authoring pedagogy.

Even if some attempts have been made for educational content to be reusable (Stash et al. 2004; Bra et al. 2006; Stash 2007), none of the above authoring tools conform to a standard such as SCORM, thus limiting their reusability. Even, when many popular LMS support SCORM courses (Moodle, Claroline, Web-CT), they do not include tools for course authoring. Hence, they support only the import of ready SCORM compliant courses developed by external SCORM authoring tools.

Table 9.1 Comparison of adaptive systems and tools

System	Adaptive Courses	Learning styles	Authoring Tool	SCORM Courses	LMS
INSPIRE	✓	Honey & Mumford			
AES-CS	✓	FD/FI			
TANGOW	✓	Felder & Silverman			
LSAS	✓	Felder & Silverman			
CS383	✓	Felder & Silverman			
EDUCE	✓	MI theory			
AHA	✓	✓	✓	Import	
WELSA	✓	ULSM	✓		
MOT 2.0	✓		✓		
VIDET	✓		✓		
REDEEM	✓		✓		
Moodle				✓	✓
Claroline				✓	✓
WebCT				✓	✓
Reload			✓	✓	
Exe			✓	✓	
Lectora			✓	✓	
Lersus			✓	✓	
ProPer & ProPer SAT 2.0	✓	Honey & Mumford	✓	✓	✓

Some of the most common SCORM authoring tools are the systems Reload and eXe, which are available for free. A drawback, however, is that even though these tools provide fast course packaging implementation, the author is still required to have good knowledge of SCORM specifications. Moreover, the educational content must be designed in an HTML editor or the author might need to invest in expensive authoring tools, such as Lectora or Lersus editors. Worse, still is the case were adaptive presentation of content to user learning style is required and authors have to write in a more or less complicated JavaScript.

Table 9.1 clearly summarizes the characteristics of both the related work and the case study systems. As shown in Table 9.1, there is no one system that implements all the characteristics that are listed above. We propose the development of a system that combines the features of an adaptive system and an LMS, is compliant to SCORM standard, supports adaptation to user learning styles and provides tools for easy authoring even by non-technical authors.

9.4 Proposed Framework

Adaptive and Intelligent Tutoring Systems (ITS) provide a personalized learning process. But the development of adaptive courses is not based on any common framework. Thus, there is a need for adaptive and ITS authoring tools which enable teachers without technical knowledge to design and author adaptive courses.

This Sect. proposes a framework that makes it possible for authors to create adaptive courses which are reusable. The proposed construction framework helps authors create courses that are adaptive to user learning style and which can be reused by many systems and platforms. The development of adaptive courses requires particular actions in specific sequence. The proposed framework focuses on courses that are adaptive to user learning style. It is based on the framework presented in Kazanidis and Satratzemi 2009b, which has been further extended to include automated adaptive courseware construction. It consists of two main layers: the Pedagogical Design Layer which is mainly concerned with the actions that the educator has to complete before the course is implemented, and the Technical Layer which refers to both the author's actions and the capabilities and features of the authoring tool. Both layers and their stages are summarized in Fig. 9.1.

9.4.1 Pedagogical Design Layer

The Pedagogical Design Layer is made up of the theoretical models on which the educators design their courses and applied instructional strategies. It includes three basic activities: 1) the educational goals of instruction have to be defined, these goals allow for the evaluation and/or measurement of the learning process; 2) the instructor has to decide on the content that is to be presented to the user, which must cover all the predefined educational goals and 3) to put into effect the definition of the applied instructional strategies.

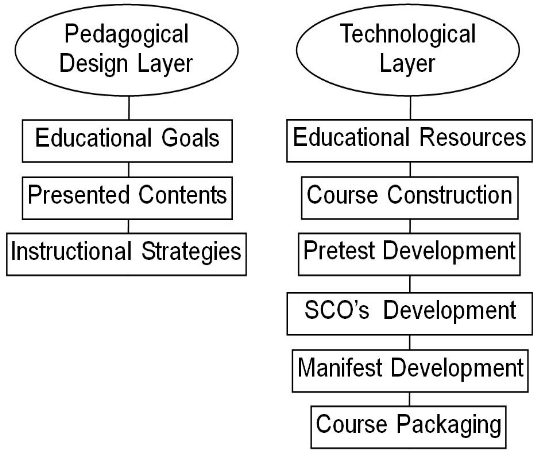


Fig. 9.1 Proposed framework layers and stages

Here, the instructor studies all the parameters, such as the student groups that have been generated according to the particular learning style, the teaching strategies for each group, the methods of acquiring user learning style, and then decides on the possible adaptation strategies that should be applied for each learning style model. At this point, the authoring tool supports a variety of widely accepted instructional and adaptation strategies and requires the teacher to choose the desired one. It must be noted that there should be an appropriate help for each of the supported instructional strategies in all the course authoring steps, so that the teacher, regardless, of his/her teaching field and programming knowledge, is able to use them in their courses. The Pedagogical Model is completed when the learning style model has been selected.

9.4.2 Technical Layer

The Technical Layer concerns the actions for implementing course construction and can be divided into six main stages. In this chapter, we adopt SCORM in order to support reusability of the educational content and intelligent adaptive functionality inside the course content code. However, with small modifications, this layer may be applied to the development of courses that are compliant with other standards, such as LOM, IMS etc.

In order to find educational resources, the user needs to find or create the necessary educational material for the course. The material should be related to the predefined course content. Additional resources may be necessary in the case that the learning style model requires the same information to be delivered through different types of media (text, graphs, videos etc), or other types of educational material (theory, activities, examples etc). Further still, a range of difficulty or even a variety of written languages could be provided to cover the different levels in user knowledge.

Developed authoring tools for adaptive courses have to give maximum support to educators in two ways: 1) support authors in writing new educational content; 2) help them discover and reuse existing educational material easily. In particular, authoring tools have to provide teachers with a simple and usable interface which includes enhanced online editors. In addition, the authoring tool interface should ask teachers to complete a form with additional data about the created educational content, the difficulty and educational levels for the educational content that has been developed to be readily accessible and easily reused by other authors. Consequently, a major objective of authoring tools should be: assist authors to discover and reuse existing quality educational material from other courses and/or by other authors. For this reason, such tools must allow authors to make educational material available to the community for sharing and reuse. In this way, it is possible to provide a list of all the educational modules to be reused and let the author choose which module and in which position they want to insert it in their course. Authoring tools automatically suggest and/or import new knowledge modules (KM) related to the existing educational material of the course.

Concerning the course construction design, the instructor designs the course structure. The course is separated into concepts and course maps; also the relations between each concept are defined. Authoring tools should provide easy and practical ways for creating and modifying the structure of each course, as well as the relationship between course concepts.

With the aim to develop a pretest in the case of adaptive SCORM compliant courses, we propose the development of a mechanism within those courses that creates a user model for every individual learner. Such a mechanism has a variety of characteristics, like user learning style, expertise level, occupation, knowledge of the course domain etc. A pretest can be taking in order for the user to ascertain the characteristic that apply to him/her, which are stored in one or more of the SCORM objectives that constitute the UM.

Most learning style models incorporate such a pretest that categorizes users into different learning style groups. Therefore, the author has to build a pretest that can discover a user's learning style and through the SCORM RTE, store it in the corresponding SCORM objective(s).

In this stage, the authoring tools should be able to automatically insert a pretest in accordance with the adaptation strategy that the author has decided to apply to the course. The pretest results should be stored in a UM so that the course content presentation and/or navigation can be adapted accordingly.

The development of learning objects should account the adopted technological standard SCORM. This stage concerns the course's SCO development. SCO must contain all the necessary educational material plus the code that will communicate with the SCORM RTE. Any learning strategy has to be applied according to the adopted learning style model. The SCO must be able to comprehend additional Javascript with the required intelligence. In particular, each SCO must read the SCORM objectives that constitute the UM and adapt their content presentation and/or sequence according to their values and therefore to the UM.

This task requires not only that the teacher put great effort into course authoring, but also that he/she has good knowledge of SCORM or any other similar technical model. This then, makes it difficult for authors without programming knowledge to create SCO manually. That is why the authoring tool needs to request for information in a simple and illustrative manner and then generate the appropriate code and append it to the educational material.

The stage of course manifest development concerns the aggregation and description of the course content into an appropriate file. In the SCORM adoption, course construction is completed by the development of the manifest file. This file includes the essential information about course structure and sequence according to CAM. Authors have to write the XML code in a editor to describe course structure and the SCO attributes. This action, requires deep knowledge of SCORM standard and XML. That is why, the authoring tools should be able to acquire all the required data and create this particular file entirely automatically.

As regards with course packaging, an authoring tool should be used for the development of an adaptive course. As a last stage, it concerns packaging all the course files and their delivering to the author in a Zoning Improvement Plan (ZIP) file. This file will be ready for use by any SCORM compliant LMS.

9.5 Case Study

Initially, in order to apply and evaluate the proposed framework, an adaptive LMS called ProPer (Kazanidis and Satratzemi 2009a) and an authoring tool, called ProPer SAT (Kazanidis and Satratzemi 2009c) were implemented. ProPer SAT is an authoring tool that is used as an external tool for ProPer, which is an adaptive learning management system.

ProPer SAT provides adaptively through both adaptive navigation and the adaptive presentation of educational content. ProPer SAT allows authors who have no or very little programming knowledge to create SCORM compliant adaptive courses. The content presentation of the course can be personalized to the user's learning style in accordance with the Honey and Mumford Model. The mechanism adopted by ProPer SAT makes it easy to add adaptation to other learning style models or user characteristics.

9.5.1 ProPer

Since ProPer SAT is an external tool of ProPer (Kazanidis and Satratzemi 2008; Kazanidis and Satratzemi 2009a). ProPer was developed as a system to provide adaptive features and support the import of SCORM compliant courses. The authoring tool ProPer SAT was developed with the intention of allowing non-technical users to take advantage of ProPer's functionality. The architecture, functionality and evaluation results of ProPer are briefly presented as follows.

ProPer is fully compliant to SCORM and its specifications, since it conforms to all SCORM requirements and supports all the necessary functions. Consequently it incorporates a combined architecture of AEHS and SCORM LMS. It involves four main modules: DM, UM, AM (from AEHS general architecture), and RTE Sequencer from SCORM LMS architecture. It takes advantage of the strengths of system categories. It is worth mentioning that in so far as a system supports specific functionalities, SCORM does not set any limitations as to its development.

Thus a compliant system, such as ProPer, is implemented by a variety of technologies, as long as it can support certain SCORM specified queries. ProPer is adapted to a learner's progress, previous knowledge, goals, navigation through the course, as well as his/her learning style. Adaptive functionality is provided in two ways: AN, using the system's Adaptation Module, and AP and/or some AN technologies, by using of SCORM functionality at the course authoring.

More specifically, ProPer applies AN through direct guidance and adaptive annotation while supporting link hiding and link disabling techniques. Direct guidance is provided with a "Next" button on the screen (Fig. 9.2) so as to deliver the most appropriate material for study according to the current UM. Adaptive annotation is applied to the links for the Table of Content (TOC).

Links are annotated for five instances: those already visited, those whose corresponding concepts are considered as known, those that constitute the learner's goals, those of the currently opened web page, and the next proposed link for study. At the level of course and main units, the system annotates whether these are considered as known and whether the user goals of each have been met. The link hiding and link disabling techniques are applied with specific SCORM conditions to the course's DM.

AP can be provided utilizing SCORM functionality. In particular, this may be applied through the following steps:

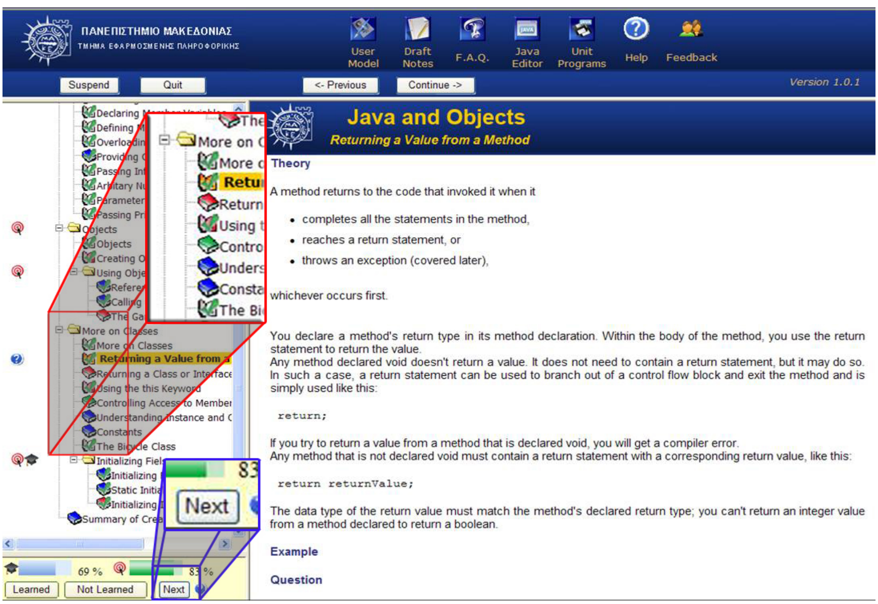


Fig. 9.2 ProPer's Adaptive Navigation Techniques

Firstly, initially a pre-test SCO acquires a user learning style either through a statement or through the analysis of a learning style questionnaire. The user learning style/preferences value can be stored in a SCORM objective, which should be global and all the other SCO courses should be able to read it.

Secondly, each SCO which delivers educational material can read the value of the corresponding SCORM objective prior to the content delivery and provide adaptive content presentation according to this objective's value. This is done through appropriate JavaScript in the HTML code of each SCO HTML file. In this way, the system can provide intelligent functionality, such as: AP that displays appropriately selected SCO content; provide (or not) "Previous" and "Continue" buttons; specify cases to hide the TOC; include the appearance of personal messages; define prerequisite concepts; change the design of the presentation.

ProPer also provides a variety of additional educational features, on the one hand, for the learner (e.g. instant feedback, dynamic Frequent Asked Questions, draft notes, Java Online Editor-Compiler, etc.) and on the other, for the teacher (course-user statistics, course management option etc.).

Both formative (Kazanidis and Satratzemi 2009a) and summative evaluation of ProPer have been carried out. For the formative evaluation, twenty two Information Technology (IT) students participated in responding to a profile questionnaire and scored on a pre-test on Java Objects. Following the 'traditional' evaluation methods of adaptive learning environments, subjects were divided into two equal groups according to their responses on the pre-test.

Group A worked on ProPer and the group B on the ADL SCORM Runtime Environment 1.3.3 which does not provide adaptation. Following, a detailed scenario of the experiment was given to the subjects and they began studying the course. The same procedure was followed for another three sessions. In the final (fifth) session, subjects in group A, who worked on ProPer, completed an assessment questionnaire about the system. Afterwards, both groups took a test on Java Objects similar to the pre-test in order to estimate the knowledge gained.

Formative evaluation results demonstrated that with ProPer students navigate in a more goal oriented manner through a course (avoiding unnecessary concepts), which results in the course being accomplished faster. T-test results found a significant difference (p=.001) in course completion between the two groups.

In the summative evaluation, 62 students were divided as in the formative evaluation, into two equal groups according to their responses on the pre-test and the learning style questionnaire. This was followed by a 2-hour session where the participants studied the course. In the final (third) session they were given another 2 hours to study the course and then in the last (third) hour the participants took the post-test which included the same questions as the pre-test.

Summative evaluation results verified that ProPer enhances the learning outcome in comparison to a simple hypermedia system. The t-test found a significant difference (p=.046, group A mean 25.41%, group B mean 17.96%) in the knowledge gained by the students that used ProPer. Likewise, users stated that they liked studying on ProPer as they found it both simple and useful. Writers also found ProPer very easy and useful for their course delivery since it allows them to design personalized instruction. At the same time, existent content can be easily retrieved, accessed and reused taking advantage of SCORM functionality.

9.5.2 ProPer SAT (SCORM Authoring Tool)

ProPer SAT exploits SCORM API and utilizes the framework presented in this paper. In a few easy steps, it allows authors to create adaptive SCORM compliant courses. More specifically, it prompts the author to fill in certain predefined HTML forms, according to the type of course, adaptive or not, that she/he wants to develop. In this way, the author needs only to choose the educational strategy of the course and to upload the educational content.

There is, thus, no need for the author to hold prior programming knowledge. Moreover, in its second version (ProPer SAT 2.0) an enhanced intelligence mechanism proposes appropriate KM to the author that could be imported to the course that is being developed. It should also be noted that ProPer SAT conforms to the above-mentioned SCORM specifications, and therefore, the delivered course packages, being fully compliant with SCORM, can be used by all the SCORM compliant platforms.

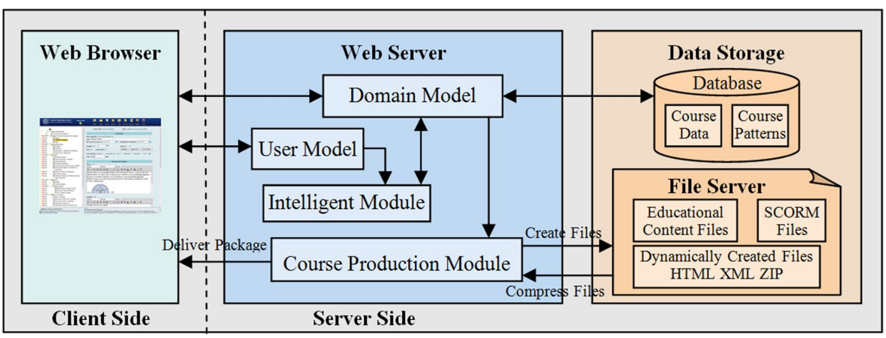


Fig. 9.3 ProPer SAT architecture

9.5.2.1 System Architecture and Implementation

The system adopts a three-tier architectural design which is identical to that set within the conceptual model (Fig. 9.3). The first tier on the client side involves the user interface. The middle tier has the system's intelligence by four modules:

1. The DM which reads data from the interface and forms and stores it onto the database,
2. The UM records user personal preferences,
3. The Intelligent Module calculates the most appropriate KM that are available for reuse and proposes their import to specific parts of the course,

4. The Course Production Module which reads data from the database and dynamically creates the user interface. It communicates with the DM and the database, interprets user preferences into the appropriate XML or HTML code and creates all the required course files. Following, it compresses these files into a ZIP archive and delivers it to the user.

Lastly, Data Storage is the back end tier and contains the system's database as well as a file server, whose files fall into three categories:

1. SCORM package files necessary for the creation of a SCORM package. SCORM standard requires the existence of some standard files in every compliant course. The produced courses are based on specific JavaScript and Cascading Style Sheets (CSS) files according to the selected type and template. These pre-exist in the system's file server and are copied in the exported SCORM course packages,
2. Educational content files uploaded to appear on course pages,
3. Dynamically created files (HTML, XML, ZIP files) as previously referred to.

The system uses an Apache Tomcat 5.5 as a web and application server and MySQL 5 as a database server. The intelligence of ProPer SAT 2.0 and its dynamic pages were implemented through Java Server Pages (JSP) and Java Servlets, while HTML, CSS and JavaScript are used for the interface implementation.

9.5.2.2 System Intelligent Functionality

The intelligent functionality of ProPer SAT 2.0 lies on two main axes: 1) smart suggestions for the import of adequate knowledge modules in a course; 2) automated suggestions for course improvement, according to their metadata and the mean assessment score of the community of authors.

In the first case, the system proposes KM from other courses or by other authors that could be imported into the current course. These proposals can either appear automatically at the course level, where the most appropriate KM as regards the entire course are displayed below the TOC or are manually presented at the KM level, whenever the author selects the button in the corresponding form. When done manually, the module is imported into a specific position in the course.

The calculation of the most appropriate KM for import is based on three criteria: 1) the content metadata, the declared keywords, the content description, the difficulty, and educational levels of the content; 2) previous author preferences and assessments on specific KM; 3) course and community of authors' assessment results. In addition, the calculation of the most appropriate content for import goes through three stages respectively.

At the first stage, the system compares the metadata of the current course's KM with those of the proposed modules for import. For every knowledge module, the system stores: a short description, up to three keywords, the difficulty and educational levels, as well as the time required for study completion. ProPer SAT 2.0 finds all the KM that have identical keywords and/or descriptions and that belong to the same educational level with those used in the course modules. Those with the highest similarity are ranked at the top of the calculated list. In the case that the suggestions concern the import to a specific part of the course, one added factor is used for more efficiency which is the level of difficulty. For example, if the KM prior to its import position has been characterized as “medium” in regards to its level of difficulty, then it is more appropriate for the following module to have a higher degree of difficulty, whereas a KM characterized as “very easy” or “easy” should not be imported to the specific position.

The second stage excludes the particular KM that the author has assessed with a low score and promotes those that have either a high score or belong to a course from where other KM where imported in the past. Finally, at the third stage of the calculation, the first n KM in the ranking list are sorted in regards to the mean assessment score of the community of authors, where N is the maximum number of displayed courses defined by the author.

The second type of proposals put forward by ProPer SAT 2.0, concern course enhancement through the delivery of a list with potential improvements for the course. This list indicates and suggests: 1) updating KM that have a low assessment score; 2) moving those KM that have the same keywords as the previous KM and which are characterized as being easier; 3) the possibility of removing consecutive KM with similar keywords, description, and difficulty level.

An example of recommendations is shown in the list (part E) at the bottom left of Fig. 9.4. Firstly, there are three items for import, which may be previewed, edited, and imported into the course by pressing on the icons on the left, respectively. If the author believes that a recommendation is not adequate for the course, it can be removed with the red “X” icon on the right-hand side of the recommendation. In addition to the above, the system provides two recommendations for course improvement. As is illustrated in the example (part D (Fig. 9.4)), the author has placed the “Advanced Learning Styles” KM before that of “Learning Styles”. The system proposes to the author to change the “Learning Styles” position, since this is the easier KM of the two. The second recommendation for course improvement is an update of “Advanced Learning Styles” content due to its having collected a low assessment score. Authors can also remove the recommendations that they disagree with.

9.5.2.3 Authoring Process

As previously mentioned, ProPer SAT provides tools so as to enable authors to develop courses that are adaptive to user learning style according to the Honey and Mumford model. Authors create multiple modules for every course unit and their presentation sequence is personalized according to user learning style. This subSect. demonstrates the authoring process of those courses as it is summarized by the sequence diagram in Fig. 9.5.

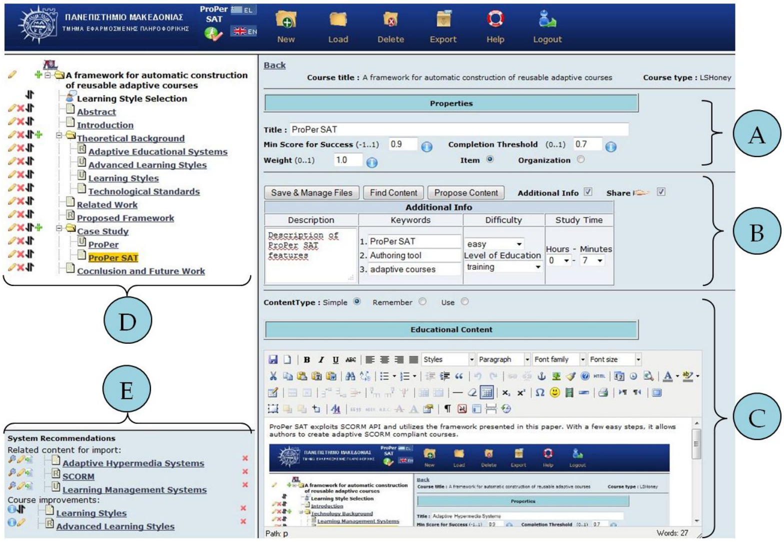


Fig. 9.4 System interface: form for content orientation and system recommendations

ProPer SAT implements and automates the framework for the development of adaptive SCORM courses. When the author creates a new course, the system asks for a course title, a description and the keywords. At the same time, it prompts author to select an instructional strategy. There are two types: The simple SCORM courses without adaptive presentation, and courses adaptive to user learning style. The latter adopts the Honey and Mumford Model implemented in ProPer and INSPIRE. Author selects one predefined template for the course interface.

In the next step, a screen with the course table of contents appears. For a simple course, only the title with the "add" option (+ symbol) appears in the tree view at the initial stage. In the adaptive type courses, however, a predefined SCO for learning style selection has been added, which allows authors to add items or organizations to their course. For every new item, the author has to define its properties or leave the default as it is and continue with content orientation through enhanced text editors (Fig. 9.4). The form is divided into three parts.

Part A contains the SCO title and its properties, such as weight, minimum score for success, and completion threshold that are essential for user progress assessment.

Part B enables the easy discovery and re-use of the SCO, as well as the import of content by other authors' and/or from other courses. The author may choose whether or not to share this KM and to provide the appropriate "Additional Info". This information is actually the KM metadata that later facilitates the discovery of the educational content.

The buttons "Find Content" and "Propose Content" that promote the re-use of existing content are very important. The first presents a screen with a list of the available KM for import and their scores according to the community assessment.

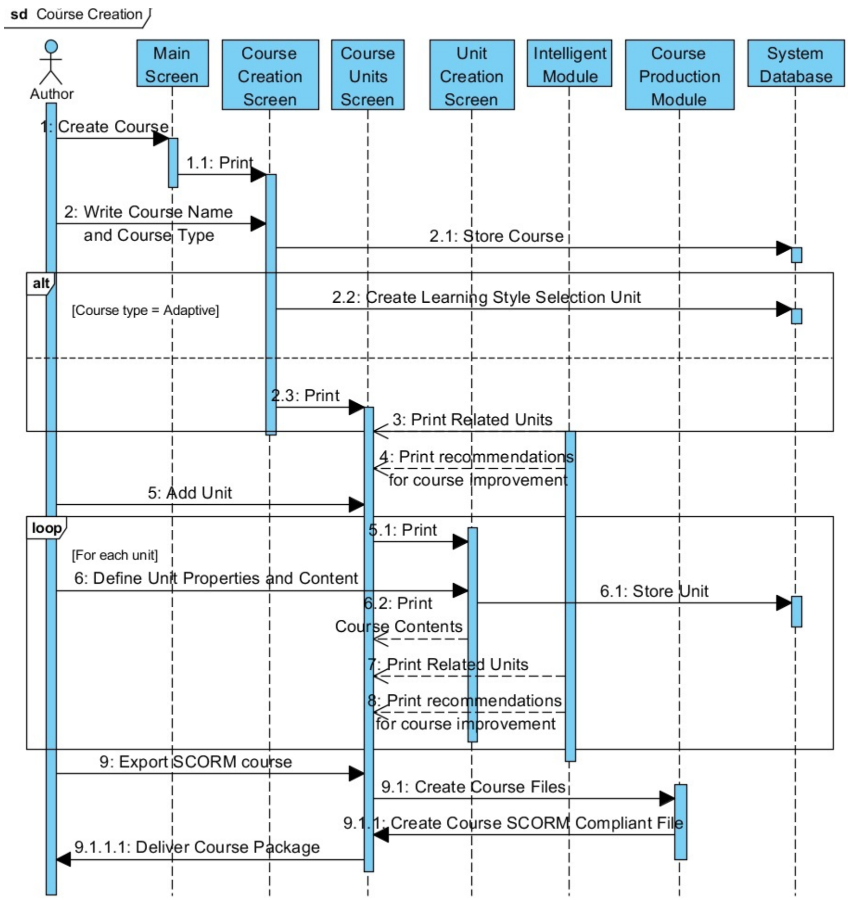


Fig. 9.5 Sequence Diagram for course creation

Each author's KM appears at the top of the list, which is followed by the list of the community KM. On this screen, the author can preview or import the available KM into his/her course. The second button prints a similar suggestion list, which according to the system's calculations, is the most appropriate KM for import in the specific course position. Similarly, the author can preview or import one of the listed items into his/her course.

Also of note is the "Save & Manage Files" button, it permits author to upload and manage images that are later inserted into the educational content of the SCO.

Part C lets the author choose the content type (Simple, Remember or Use) to compose new educational content through an enhanced HTML What You See Is What You Get (WYSIWUG) editor. The content type Simple presents content statically, whereas Remember and Use are adaptive content types. These types allow the user to create content for different KM (Theory, Example, Question, Activity, and Exercise) within a SCO. These modules are presented in a different order, according to user learning style. For example, if a student has been characterized as an Activist, according to the Honey and Mumford model, it is better to follow an instructional strategy that initially requires the student to answer a question in the specific domain and then provide him/her with an example module so that the (Activist) student can find out how this knowledge is applied. The system provides the student with a theory hint which presents the domain theory. On the other hand, a Reflector student learns better if she/he starts her/his study with Theory and continues with an Example and a Question. More details on applied adaptivity are presented in the next subsect.

When the author finishes with content orientation she/he can save it and create another. Additionally, from the tree view of the TOC (Fig. 9.4 part D), the author can update, delete or move an existing KM; while in the bottom left part of the screen (Fig. 9.4 part E), she/he can check a list with the most appropriate KM to import into the course. Authors are able to preview, assess, and import these KM in a position in the course they prefer. They read and apply system recommendations for course improvements, like those in the example already presented.

Finally, when course construction is completed, the author can choose to export it through the respective button at the top of the screen. In this case, ProPer SAT automatically creates course files and packages them, according to SCORM specifications in an appropriate ZIP archive.

This file contains the course manifest file, HTML files and other files necessary for SCORM compliant courses. In order for the system to develop these files, it uses a common, stable code for every SCO of the same type (Simple, Remember, Use), as well as the material that the user has imported and stored in the database. For the present, courses can be exported to SCORM 2004 3rd and 4th editions which enable this file to be imported into ProPer and all the SCORM compliant LMS.

9.5.2.4 Adaptive Courses in ProPer SAT

The adaptive courses that ProPer SAT automatically develops are based on a case study presented in Kazanidis and Satratzemi 2009b. More specifically, following the framework used by both INSPIRE and ProPer, we present the same content to the user with emphasis on specific knowledge modules in keeping with the user's learning style. In order to achieve this, SCO with various knowledge modules (theory, example, question, activity, etc) that are presented on one page, had to be created. The module presentation sequence depended on the user's learning style. For example, if the user is categorized as an 'Activist', then the presentation of a Remember type of educational material would start with a Question followed by an Example and a Theory hint. Similarly, a 'Reflector' starts his/her study with Theory and continues with an Example and a Question and so on.

On completion of course development, the system generates the course HTML files which have to be enhanced with the appropriate code for adaptivity. Fig. 9.6 shows the code of an HTML page corresponding to a Remember type SCO. The words in bold and italics correspond to data that had been imported by the user, stored in the database and have been retrieved from there.

More specifically, the code in Part A calls up certain JavaScript files that in turn contain all of the appropriate code for communication with the system and adaptation to user learning style. Part B of the code is the "body" tag of the HTML file and summons a JavaScript function responsible for the initialization of the SCO, as well as the sequence and appearance of the SCO knowledge modules, according to user learning style. The Boolean parameters of this function specify whether or not the educational content for the theory, example, and question knowledge modules exist in this SCO. Part C of the code calls up a function that loads the title of the page as this is retrieved from the system's database. Lastly, the code in Part D involves three blocks, each of which contains the educational material for the theory, example, and question knowledge modules, respectively.

The educational material of these blocks is retrieved from the system's database. For example, a learning unit with the title "ProPer SAT Functionality" that includes Theory and Example knowledge modules will set the code into the bracket of Part B (Fig. 9.6) to "true, true, false" and the code into the bracket of Part C to "ProPer SAT Functionality". Part D of the code is included automatically, according to the data inserted in the Theory and Example knowledge modules of the learning unit, respectively; while, the Question code will be left blank since this learning unit does not include a question knowledge module.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
   <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
   <link rel="stylesheet" href="../css/style.css" type="text/css" />
   <script type="text/javascript" src="../scripts/snceFunctions.js"></script>      A
   <script type="text/javascript" src="../scripts/APIWrapper.js"></script>
   <script type="text/javascript" src="../scripts/objectivesFunctions.js"></script>
</head>
<body  onload="loadLSRememberSubPage(true,true,true)"      B
onunload="unloadPage()">
<div id="wrapper">
   <script type="text/javascript" language="javascript">
         loadThisHeader("Title of the SCO");      C
   </script>
   <!-- Main page content -->
    <div id="fullScreen">
      <div id="content">
        <script type="text/javascript" language="javascript">
            loadLSRememberBody();
        </script>
        <div id="theory" style="display:block;text-align:justify">
                    Theory educational material
        </div><p>
        <div id="example" style="display:block; text-align:justify">
                Example educational material      D
         </div> </p><p>
         <div id="questionContent" style="display:block;text-align:justify">
                Question educational material
         </div></p>
       </div><!-- end  content -->
     </div> <!-- end fullScreen -->
</div> <!-- end wraper -->
</body>
</html>
```

Fig. 9.6 Generated Remember Type SCO HTML file

9.5.2.5 System Evaluation

ProPer SAT includes a number of features and tools that have been tested in light of the specific evaluation goals. To test the extent of acceptance by users, the Technology Acceptance Model (TAM) (Davis 1989) was applied. According to TAM, the adoption of a particular type of technology mainly depends on its perceived ease of use and usefulness, and should, thus, be carefully evaluated.

The formative evaluation procedure was divided in accordance with the Tessmer Model (Tessmer 1993) into the following parts: Experts review, one-to-one evaluation, and field trial.

An appropriate questionnaire to evaluate the system was designed, which was divided into four sections. The first section determines teachers' opinions about system functionality.

The second and third sections record teachers' responses to system usability – ease of use and usefulness in accordance with the TAM model and Nielsen Heuristics. The final section allows teachers to make additional recommendations, comment on what they like or dislike most and indicate possible bugs or confusing parts in the system. Five-point Likert scale questions were used in the first three sections of the questionnaire.

An experts' review was conducted through a semi-structured questionnaire-based interview, in order to record opinions about ProPer SAT design and functionality. Three experts, whose educational background was in computer science acted as evaluators. Overall, the feedback they gave was positive, finding the system both easy to use and very useful.

Nevertheless, it might be worth mentioning that a couple of points were brought to our attention. Firstly, that it would be even more useful for non-technical users to be able to access added information about the Honey and Mumford Model and how this can be applied to the courses, which were promptly complied with. Secondly, why the system does not support SCORM conditions and rules, which is for reasons of simplicity, however, it should be noted here that research is underway to apply them into the courses.

The aspect that the experts claimed to like most was the system's intelligence mechanism of automated suggestions for the import of identical content. They tested the system's efficiency to propose the appropriate KM for import by initially creating the corresponding KM in one course and checking whether the system would suggest the right KM in an identical course. The system was found to be highly efficient, in spite of the fact that authors are only able to exploit this feature further when the system's knowledge pool contains a large range of KM by different authors or from different courses.

The next stage of the formative evaluation was the one-to-one assessment, which had a similar procedure to that of the experts' review (i.e., first the system was introduced and then focus was placed on its functionality). A semi-structured interview followed along the lines of the predefined questionnaire. Four teachers, two from secondary and two from tertiary education, evaluated the system. Generally, they all found the system to be both simple and useful and stated their intent to use it in the future. In more detail, one teacher, an expert in Moodle, found the system most satisfactory to use; another, who had participated in the evaluation of

ProPer, stated that ProPer SAT was the complementary tool needed; the remaining two teachers, despite not having any experience as authors of online courses, succeeded in producing an adaptive course in a few minutes. However, there was one request for the creation of more educational material for the development of adaptive courses. This was complied with by providing added functionality for educational content reusability.

The aspect of ProPer SAT 2.0 that teachers seemed to like the most was the automated creation of adaptive courses. They expressed that they would like to use adaptive courses in their classrooms to check whether and to what extent the learning outcome has improved.

In addition, ability to reuse the educational material was considered to be extremely helpful in saving time by decreasing the course construction workload. Finally, some comments were made on the issue of copyright regarding educational material stored in the system. Following the iteration of the system, the option that allows each author to choose whether or not to share the educational material they produce was added.

The goal of the field trial of the formative evaluation was to discover in more depth how easy and useful teachers considered the system to be, the usefulness of its functionality, and for them to record any possible suggestions for future work.

Thirty (30) secondary education teachers evaluated ProPer SAT 2.0. Twenty (20) were IT teachers, and the other ten (10) taught various high school subjects. The teachers where introduced to the system and the adopted adaptation theory.

Afterwards, they were asked to develop one adaptive and one simple SCORM compliant course applying three different ways: Composing new content, reusing their previously developed content, and using other teachers' content. In addition, they were encouraged to share their educational content with other teachers in order for it to be reused, assess other teachers' content which was used in their courses, and check system automated suggestions for course authoring and improvement. On having performed these tasks, they were asked to complete the predefined attitude questionnaire.

Table 9.2 shows the questions of the first section along with the variance of the results and the mean value of teachers' responses on the scale of 1 (not at all useful) to 5 (very useful). The scale was kept numerical for interval processing.

As the Table 9.2 shows, the mean of all the questions in the first section was above 3.5. This leads to the conclusion that the teachers considered ProPer SAT 2.0 and its features useful for the development of quality online SCORM courses. They found both simple and adaptive course authoring useful (Q1, Q2). Similarly, they very much liked the reuse of educational content (Q3, Q4) which ProPer SAT 2.0 supports, preferring the possibility of reusing their own material most. They believed that choosing whether or not content is available for use by other authors is practical/effective (Q5) but they seemed to have had doubts on assessment efficiency (Q6), possibly due to a lack of trust in other people's choices.

Regarding the system's intelligent features, teachers liked the intelligent authoring process that recommends related KMs (Q7). They also considered the system recommendations for course improvement (Q8). These results are in agreement with the outcomes of MOT evaluation (Stash et al. 2004) which similarly adapts authoring by recommending related content.

The second and third sections of the questionnaire examined system ease of use and usability, respectively. Overall, teachers found the system easy (3.92), and useful (4.17). In particular, teachers needed only a short time to become familiar with the system's features (3.92), and found it easy to use (4.04).

However, it appears they had some doubts as to whether the system is appropriate for users with no previous experience in computers (3.46). But, they believed that a basic knowledge of Internet applications is sufficient for problem-free use of the system.

Table 9.2 Evaluation of ProPer SAT 2.0 functionality

What do you think about …	Mean (1-5)	Variance
Q1. Simple course authoring?	4.12	0.59
Q2. Adaptive course authoring?	4	0.72
Q3. Reusing your own learning objects?	3.96	0.76
Q4. Reusing other authors learning objects?	3.92	0.71
Q5. Allowing authors to share or not their content with other authors?	3.77	0.90
Q6. Assessing other authors learning objects?	3.81	0.72
Q7. Intelligent authoring by recommending related knowledge modules?	4	0.56
Q8. Recommending suggestions for course improvement?	3.88	0.67

Concerning the questions related to Nielsen heuristics, teachers found system usability to be high (3.83). It seems that the system's help needs enhancement (3.38). Most of the above results are taken from questions with variance below 1, where the opinions were strong and with minimal spread. A *t*-test was used to check for possible differences between the responses of IT teachers and those of other subjects. No significant differences were found. This lends support to the claim that ProPer SAT 2.0 is easy for both experienced (IT teachers) and novice users, even those with no or little programming knowledge.

Qualitative results of the fourth section of the questionnaire show teachers' opinions in regards to the best and worst system features, as well as possible additions that could be implemented. In particular, the system features that teachers liked most were: system ease of use, rapid course development, the reusability of knowledge modules, and the automatic recommendations for improvements on the course structure. This confirms that ProPer SAT 2.0 is a useful and easy system for the development of SCORM courses. There were, however, some system features that were confusing to teachers. More help was required for the following: 1) specific SCORM attributes used in the KM creation, such as minimum score for success and completion threshold values; 2) content types and their use in an on-line course; 3) more explanations on the adopted learning style model.

ProPer SAT 2.0 was revised in accordance with the field test evaluation outcomes. More help on system features was added and SCORM attributes were further explained. Two bugs that teachers referred to were fixed and system design was further simplified. The pilot study of ProPer SAT 2.0 showed that most system features, including the intelligent ones, were perceived as being useful in the creation of SCORM compliant courses. In addition, the system was perceived as being easy to use. According to TAM, these findings indicate a high level of users' behavioral intention to apply the system, which in turn leads to high actual use.

9.6 Conclusion

The proposed framework helps authors create adaptive user learning style courses which can be reused by many systems and platforms, and additionally, guides system developers to build authoring tools that are simple, appropriate for non-programmer teachers and to include intelligent functionality.

The benefits of such a framework application are: 1) easy and rapid course construction; 2) the development of qualitative courses; 3) better learner performance through a personalized learning experience; 4) reusability of educational content.

Two systems were developed as a case study. ProPer delivers SCORM compliant adaptive courses that may, among other things, adapt content presentation according to user learning style. The system adapts to the learner's characteristics, preferences, goals and learning style. ProPer's main advantage over similar systems is that it succeeds both in providing adaptivity and at the same time, it adopts a standard for reusability and interoperability of the learning content.

ProPer SAT 2.0, in addition to other authoring tools for adaptive courses like AHA!, WELSA, MOT etc., is appropriate even for teachers without programming knowledge. Similar to ProPer SAT 2.0 are REDEEM and VIDET, however, unlike our system, these systems do not focus on the reuse of educational content even when some attempts for reusable educational content have been made (Stash et al. 2004), and since these authoring tools do not conform to a standard such as SCORM, they have limited reusability. Furthermore, ProPer SAT 2.0 incorporates intelligent functionalities, such as automatic suggestions for content import and for course improvement. These functionalities help authors to enhance their courses with relevant qualitative material and be able to further improve them should an issue of wrong design be detected.

ProPer SAT 2.0 and ProPer together can create and deliver courses adaptive to user learning style in the same way that AES-CS and INSPIRE do, thus providing a tool for quick course production. In sum, ProPer and ProPer SAT 2.0 offer a promising solution for non-technical teachers who want a simple way to be able to create and deliver SCORM compliant and/or adaptive courses quickly and easily.

Nevertheless, there are a few limitations to the current research. The proposed framework is based on the adoption of a technological standard such as SCORM. This option enables content reusability and interoperability; however, the courses produced have to fulfill its specifications. How a variety of learning activities, such as collaboration learning may be applied to courses that adopt specific technological standards requires further study. In addition, ProPer SAT 2.0 fulfills web 2.0 requirements were users act as a community and share, create, comment and assess educational content. For this reason, ProPer SAT 2.0 functionality needs to be enhanced with more tools that permit authors to cooperate on the development of educational content.

Last but not least, investigation needs to be made into the active participation of students that use ProPer in a way that enables them to change and comment on course contents participation of students that use ProPer in a way that enables them to change and comment on course contents.

Acknowledgments. This work was funded by the Research Committee of the University of Macedonia as part of the Basic Research Project.

References

ADL. SCORM, `http://www.adlnet.gov/Technologies/scorm` (accessed April 29, 2011)

Ainsworth, S.E., Major, N., Grimshaw, S.K., Hayes, M., Underwood, J.D., Williams, B., Wood, D.J.: REDEEM: Simple intelligent tutoring systems from usable tools. In: Murray, T., Blessing, S., Ainsworth, S. (eds.) Authoring Tools for Advanced Technology Learning Environment, pp. 205–232. Kluwer Academic Publishers, The Netherlands (2003)

Armani, J.: VIDET: A visual authoring tool for adaptive websites tailored to non-programmer teachers. Educational Technology & Society 8(3), 36–52 (2005)

Bajraktarevic, N., Hall, W., Fullick, P.: Incorporating learning styles in hypermedia environment: Empirical evaluation. In: Proceedings of AH Workshop (2003)

De Bra, P., Smits, D., Stash, N.: Creating and Delivering Adaptive Courses with AHA! In: Nejdl, W., Tochtermann, K. (eds.) EC-TEL 2006. LNCS, vol. 4227, pp. 21–33. Springer, Heidelberg (2006)

Bradley, W.E.: Conceptual framework for the design and evaluation of online learning modules in professional training and academic education in business. The Business Review, Cambridge 18(1), 196–207 (2011)

Brusilovsky, P.: Adaptive hypermedia. User Modeling and User Adapted Interaction 11(1/2), 87–110 (2001)

Carro, R.M., Pulido, E., Rodríguez, P.: Dynamic generation of adaptive Internet-based courses. Journal of Network and Computer Applications 22, 249–257 (1999)

Carver, C.A., Howard, R.A., Lavelle, E.: Enhancing student learning by incorporating learning styles into adaptive hypermedia. In: Proceedings of ED-MEDIA, pp. 118–123 (1996)

Dagger, D., Wade, V., Conlan, O.: A framework for developing adaptive personalized e-Learning. In: Nall, J., Robson, R. (eds.) Proceedings of E-LEARN, pp. 2579–2587 (2004)

Davis, F.D.: Perceived usefulness, perceived ease of use, and user acceptance of information technology. MIS Quarterly 13(3), 319–340 (1989)

Duval, E.: Standarized metadata for education: A status report. In: Montgomerie, C., Karmo, V. (eds.) Proceedings of ED-MEDIA, pp. 458–463 (2001)

Felder, R.M., Silverman, L.K.: Learning styles and teaching styles in engineering education. Engineering Education 78(7), 674–681 (1988)

Gardner, H.: Multiple intelligences: The theory in practice. Basic Books, New York (1993)

Ghali, F., Cristea, A.I., Stewart, C.: My online teacher 2.0. In: Proceedings of Individual and Group Adaptation in Collaborative Learning Environments Workshop of ECTEL (2008)

Honey, P., Mumford, A.: The manual of learning styles. Peter Honey Publications, Maidenhead (1992)

Kazanidis, I., Satratzemi, M.: Adaptivity in a SCORM Compliant Adaptive Educational Hypermedia System. In: Leung, H., Li, F., Lau, R., Li, Q. (eds.) ICWL 2007. LNCS, vol. 4823, pp. 196–206. Springer, Heidelberg (2008)

Kazanidis, I., Satratzemi, M.: Adaptivity in pro per: An adaptive SCORM compliant LMS. Journal of Distance Education Technologies 7(2), 44–62 (2009a)

Kazanidis, I., Satratzemi, M.: Applying learning styles to SCORM compliant courses. In: Aedo, I., Chen, N., Kinshuk, Sampson, D., Zaitseva, L. (eds.) Proceedings of IEEE-ICALT, pp. 147–151. IEEE Press, New York (2009b)

Kazanidis, I., Satratzemi, M.: Efficient Authoring of SCORM Courseware Adapted to User Learning Style: The Case of ProPer SAT. In: Spaniol, M., Li, Q., Klamma, R., Lau, R.W.H. (eds.) ICWL 2009. LNCS, vol. 5686, pp. 196–205. Springer, Heidelberg (2009c)

Kelly, D., Tangney, B.: Adapting to intelligence profile in an adaptive educational system. Interacting with Computers 18, 385–409 (2006)

Kolb, D.: Experiential Learning: experience as the source of learning and development. Prentice-Hall, New Jersey (1984)

Krull, G.E., Mallinson, B.J., Sewry, D.A.: Describing online learning content to facilitate resource discovery and sharing: The development of the RU LOM Core. Journal of Computer Assisted Learning 22(3), 172–181 (2006)

Papanikolaou, K.A., Grigoriadou, M., Magoulas, G.D., Kornilakis, H.: Towards new forms of knowledge communication: The adaptive dimension of a Web-based learning environment. Computers & Education 39, 333–360 (2002)

Peng, Y.: SCORM-based courseware design for the personalization of e-learning system. In: Proceedings of Intelligent Information Technology Application Workshop of IEEE Computer Society (2007)

Popescu, E.: An Artificial Intelligence Course Used to Investigate Students' Learning Style. In: Li, F., Zhao, J., Shih, T.K., Lau, R., Li, Q., McLeod, D. (eds.) ICWL 2008. LNCS, vol. 5145, pp. 122–131. Springer, Heidelberg (2008)

Romero, C., Rider, J.J., Ventura, S., Hervas: AHA! meets SCORM. In: Proceedings of MCCSIS, pp. 95–98. IADIS Press, Lisbon (2005)

Stash, N.: Incorporating cognitive/learning styles in a general-purpose adaptive hypermedia system. PhD dissertation, Eindhoven University of Technology (2007)

Stash, N., Cristea, A., De Bra, P.: Authoring of learning styles in adaptive hypermedia: Problems and solutions. In: Proceedings of WWW Alt., pp. 114–123. ACM Press, New York (2004)

Tessmer, M.: Planning and conducting formative evaluations: Improving the quality of education and training. Kogan Page, London (1993)

Triantafillou, E., Pomportsis, A., Demetriadis, S.: The design and the formative evaluation of an adaptive educational system based on cognitive styles. Computers & Education 41(1), 87–103 (2003)

Tseng, S.S., Su, J.M., Hwang, G.J., Hwang, G.H., Tsai, C.C., Tsai, C.J.: An object-oriented course framework for developing adaptive learning systems. Educational Technology & Society 11(2), 171–191 (2008)

Witkin, H., Moore, C., Goodenough, D., Cox, P.: Field-dependent and field-independent cognitive styles and their educational implications. Review of Educational Research 47(1), 1–64 (1977)

Abbreviations

AC	Abstract Conceptualization
ADL	Advanced Distributed Learning
AEHS	Educational Hypermedia Systems
AE	Active Experimentation
AICC	Aviation Industry Computer-based training Committee
AM	Adaptation Module
AN	Adaptive Navigation
AP	Adaptive Presentation
API	Application Program Interface
CAM	Content Aggregation Model
CE	Concrete Experience
CSS	Cascading Style Sheets
DM	Domain Model
IMS	Instructional Management Systems
IT	Information Technology
JSP	Java Server Pages
KMs	Knowledge Modules
LMS	Learning Management System
LOM	Learning Object Metadata
MI	Multiple Intelligences
RAID	Reusable Accessible Interoperable Durable
RO	Reflective Observation
RTE	Run-Time Environment
SAT	SCORM Authoring Tool
SCORM	Sharable Content Object Reference Model
SCOs	Sharable Content Objects
SSN	Simple Sequencing and Navigation
TAM	Technology Acceptance Model
TOC	Table Of Contents
ULSM	Unified Learning Style Model
UM	User Model
WYSIWYG	What You See Is What You Get
XML	eXtensible Markup Language
ZIP	Zoning Improvement Plan

Chapter 10
Interoperable Intelligent Tutoring Systems as SCORM Learning Objects

Gustavo Soares Santos[1] and Joaquim Jorge[1,2]

[1] Department of Computer Science and Engineering, Instituto Superior Técnico,
Technical University of Lisbon,
Av. Rovisco Pais 1, 1049-001 Lisbon, Portugal
{gustavossantos,jorgej}@ist.utl.pt
[2] INESC-ID,
Rua Alves Redol 9, 1000-029 Lisbon, Portugal
jaj@inesc.pt

Abstract. Learning technologies are currently present in many educational institutions around the world. Learning Management Systems (LMS), Personal Learning Environments (PLE) and other types of educational platforms are very popular and now common in our schools and universities. However, most of the educational content currently available in the educational platforms is non-adaptive and non-intelligent educational content such as HTML pages, PDF files and Power Point Presentations (PPT). This type of content does not provide the high quality educational assistance that technology can provide. On the other hand, intelligent and adaptive educational systems are a successful and mature field of learning technologies that can provide very high quality educational assistance. In order to allow Intelligent Tutoring systems (ITS) to be loaded into different types of educational systems, we have developed an approach based on E-Learning standards. Our approach is also grounded in a very well known paradigm for implementing ITS, and the main goal of this chapter is to present a novel approach for implementing ITS as learning objects using the Sharable Content Object Reference Model (SCORM).

10.1 Introduction

Learning technologies and educational systems are now part of the infrastructure in many educational institutions around the world. LMS, PLE and other types of educational platforms are now very common in our schools and universities (Beatty and Ulasewicz 2006). Unfortunately, these educational tools have been mainly used to store plain educational content (Sabbir-Ahmed 2004). This type of content (such as PDF and PPT) cannot provide the high quality educational assistance that technology can (Brusilovsky et al. 2007).

On the other hand, adaptive and personalized educational systems can provide very high quality educational assistance. For instance, ITS are adaptive

A. Peña-Ayala (Ed.): Intelligent and Adaptive ELS, SIST 17, pp. 239–265.
springerlink.com

educational tools that offer direct personalized instruction and feedback to students (using artificial intelligence techniques, cognitive psychology and learning sciences). ITS have been used in several domains, from middle school math (Ritter et al. 2007) and physics (Vanlehn et al. 2005), to programming languages (Corbett and Anderson 1992) and military applications (McCarthy 2008). Many experiments have proved that ITS can be beneficial to learning (Ritter et al. 2007; Vanlehn et al. 2005; Corbett and Anderson 1992). However, their popularity outside the academia is relatively low.

Some of the main reasons for the reduced attractiveness of ITS include: 1) the intrinsic complexity of their development process (Aleven et al. 2009); 2) the impossibility of loading them in different platforms (Rey-López et al. 2008); 3) the extra effort necessary to make them available over the Web (Wijekumarr et al. 2003; Mia 1997). To address some of the limitations mentioned above, we have developed an approach (Santos and Figueira 2010a) and also prototype (Santos and Figueira 2010b) for making ITS more viable to educational institutions. Our approach allows the implementation of Web-Based interoperable ITS, and it is based on the SCORM e-learning standards to implement Learning Objects (LO) (Santos and Figueira 2011).

The main objective of this chapter is to present a novel approach for implementing ITS as learning objects using SCORM (Santos and Figueira 2010a), and also describes the intelligent and adaptive attributes of the tutoring systems that are designed with our approach. In addition, a small geometry ITS (Santos and Figueira 2010b) will be described. Our main contribution here is a method that allows solving some important limitations of ITS, providing a way to develop open source, intelligent and adaptive learning objects that can be downloaded from repositories of educational content, and loaded into different learning platforms.

10.2 Background and Related Work

The use of technology for learning and training started in the early 1940s. American researchers from Bell Labs and the US Navy developed flight simulators, where the user inputs were processed by a computer, and feedback was generated for pilot training (White 2006). In this early phase of the learning technologies the educational software was directly tied to the hardware on which it ran (usually mainframe computers).

The development of the Personal Computer (PC) in the 1970s, had some implications for educational software. Before the invention of the PC, learners depended on government or university owned mainframe computers to have access to educational software. This would involve going to a place where the terminals were located and respect timesharing limits.

After the PC shift, students could have access to educational software even at home and with different and timesharing restrictions. During the 1980s, some educational titles for PC were developed by companies like MECC, the Learning Company, and the Minnesota Educational Computing. At that time, these and some other companies specialized on the development of educational software. Later on in the 1990s, with the new advances in computer hardware (CD-ROM, multimedia graphics and sound) and the internet, major developments in educational software came true.

With the internet spreading in the second half of the 1990s, Web-sites became a very popular method for delivering educational content. Currently, Web-Based LMS like Moodle and Blackboard are some of the main platforms of delivery. With the great accessibility that the internet can provide, the field of E-learning has developed a lot. Currently, students have fewer restrictions when it comes to location and time constraints.

The main purpose of this section is to describe a few of the most significant fields of learning technologies nowadays. We are going to give special attention to topics that are important for understanding our approach (like ITS and SCORM). The first subsection will address the topics related to E-learning. The second subsection will introduce the SCORM E-learning standards. The third subsection will describe Learning Objects, and the last subsection will concentrate on ITS.

10.2.1 E-Learning, LMS and PLE

The concept of E-learning can be defined in a broader sense as the usage of technology to facilitate the learning process anywhere and anytime. E-Learning systems usually can provide training, assessment, delivery of educational content and educational guidance (Watkins 2010). One of the main purposes of E-learning is to use the power of technology to overcome the limitations of distance, time, and resources.

With the introduction of E-Learning systems in educational institutions, it is now possible to facilitate learning even when students are not at school, or have time constraints. As generally people learn in different ways, and different subjects may require different teaching strategies, teaching an E-learning course might require different E-learning delivery methods.

The main delivery methods for E-learning are synchronous and asynchronous (Hrastinski 2008). As examples of asynchronous E-Learning methods we have self-paced courses and discussion groups. Virtual classrooms and shared whiteboard are examples of synchronous E-Learning. In addition to an E-Learning delivery method, teaching an E-learning course also requires a way of publishing and managing the course.

A LMS is a computer framework that generally does the management and delivery of courses. LMS allows publishing courses and placing them in catalogues that are usually available online.

Students can therefore be assigned to courses or access the LMS to register in a course. In addition to that, LMS also provide a set of services to assist the administration of courses.

In general, the services provided by LMS are, central administration of courses, reports (about courses and students), assemble and delivery of educational content, portability of content (usually via standards), reusability of content (usually via standards), customization of content and assessment of students. Additionally, LMS can personalize instruction and track the students' activities while they are working on tasks.

Services like tracking student´s progress and personalizing instruction often require the LMS to be compliant with some standard. Standards bring order to systems and unify the way systems operate and interact. E-learning standards provides among other things, interoperability, reusability, durability, and accessibility of E-learning content. The most used E-learning standard (that is currently the "de facto standard" for E-learning and LMS) is the SCORM collection of standards and specifications. The next section of this chapter will address the SCORM.

PLE are more recent educational tools (compared to LMS). The term PLE have first been mentioned in 2004. PLE are able to support communities and services in educational platforms that students use to direct their own learning and pursue their own educational goals (van Harmelen 2008). PLE support a learner-centred view and they differ from LMS which are based on course-centred view. In general, PLE are described as systems that help students to take control of their own learning. In addition, social interaction with other learners is heavily stimulated.

Technically speaking, a PLE consists on integrating of some "Web 2.0" technologies around an independent learner. It is not a corporate or institutional application, but a personal learning hub where content can be reused and mixed according to the student's own needs and interests. A PLE is not a single application, but a collection of applications interoperating. Therefore, a PLE is consequently an environment rather than a system (Downes 2005).

10.2.2 SCORM Standards

Standards can be defined as a set of norms and requirements for technical systems. They are usually characterized by a formal document establishing uniform engineering criteria, processes, methods and practices. Standards can be developed unilaterally (for example, by a regulatory body), or they can be developed by groups (such as trade associations and trade unions).

The SCORM (ADL 2011) is a compilation of standards and specifications for E-learning. It is developed and maintained by the Advanced Distributed Learning (ADL) initiative (ADL 2010). Despite the fact that it was developed by ADL, SCORM is actually a product of several entities. Combining the work of these entities we have therefore the three main specifications of SCORM.

The three specifications that compose the SCORM model are 1) the Content Packaging specification; 2) the Run-Time Environment specification; 3) the Sequencing and Navigation specification. These specifications together define how educational content should be packaged and described, how educational content should communicate, and how educational content should be sequenced and navigated. The main benefits that come from using these specifications are interoperability, reusability, durability and accessibility (ADL 2011).

10.2.2.1 SCORM Content Packaging

The SCORM Content Aggregation Model (SCORM CAM) describes how to pack the necessary components of instruction to form an educational resource for a specific learning experience. The main objective of the SCORM CAM is to allow the exchange of educational resources between systems.

The most important part of the content packaging specification is the SCORM manifest, which is an Extensible Markup Language (XML) file that completely describes the content of a SCORM course. The manifest contains several pieces for representing a course structure.

10.2.2.2 SCORM Run-Time Environment

The SCORM run-time environment (RTE) specification describes how systems should launch the educational content, and how the content should communicate with the host system. All the SCORM communication happens after a LO is delivered to the user interface. In SCORM, a Sharable Content Object (SCO) is a LO that communicates to a LMS (sending and requesting user information).

Components of instruction that do not communicate with the LMS are called Assets (for example, power point presentations or PDF). The navigation among SCO and Assets in SCORM is managed by the SN specification that will be explained later on in this document.

10.2.2.3 Launching Content

By default, SCORM content has to be web-deliverable and communicate within the context of a Web browser session. The user interface can launch only one SCO or Asset at a time. SCORM does not specify many formal requirements for the user interface. Consequently, every LMS is a bit different when it comes to presenting content. On the other hand, we can always expect that every LMS will provide some sort of navigable menu of contents, as well as controls for flow navigation (next, previous, etc.). LMS have only two ways of launching SCO and Assets. They can be launched in a frameset, or in a new window.

In general, when a course contains only one SCO the SCO is launched in a popup window. On the other hand, when a course has many SCO, then the LMS will launch the SCO in a frameset with navigational elements.

10.2.2.4 The SCORM RTE API

After a SCO is launched in the user interface, the communication between SCO and LMS occurs only through an ECMAScript (JavaScript) Application Programming Interface API. SCORM provides an API that will therefore regulate the way SCO can communicate with the educational platforms. SCO should not communicate through other ways like form posts, web services, database writes or anything else. However, if there is really a need for, it is allowed. On the other hand, we have to stress that using external resources or services compromises the portability of the educational content (Software 2011), and therefore it is strongly not recommended. When SCO are loaded in the interface, they should contain in their code a JavaScript function to find the location of the SCORM API that has to be placed somewhere in the educational platform.

Once a SCO has found the API, it can communicate with the LMS. Only the SCO can therefore initiate the communication process. The educational platforms are passive and simply respond to the API calls. The SCORM 1.3 (also known as SCORM 2004) API has the following functions and signatures:

- Bool: Initialize(),
- Bool: Terminate(),
- String: GetValue(CMIElement : element),
- String: SetValue(CMIElement : element, string : value),
- Bool: Commit(),
- CMIErrorCode: GetLastError(),
- String: GetErrorString(CMIErrorCode : errorCode),
- String: GetDiagnostic(CMIErrorCode : errocCode).

Basically, every time a SCO is launched and wants to start communicating, the "Initialize" function must be called first. After finishing the communication, the "Terminate" function must be called to end the communication. The other functions are basically called to manipulate the SCORM database, which contains the student model and also a set of an auxiliary data.

10.2.2.5 Sequencing and Navigation

Sequencing is a SCORM process that occurs in every time a student starts a course, terminates a SCO, or requests for a component of instruction using a navigation widget. The process of sequencing can be controlled by the Sequencing and Navigation (SN) module.

SN decides therefore what navigational controls will be available for the student, and what SCO could be delivered next. Moreover, SN orchestrates the flow of a course entirety, but it does not affect on how SCO operate and navigate internally. This is completely up to the instructional designer. In addition, the usage of sequencing in SCORM applications is completely optional. Most of the SCORM courses do not actually use any sequencing constructs.

The constructs associated with every SCORM activity that uses SN are: 1) the Tracking Data; 2) the Sequencing Definition. The Tracking Data stores information about the state of an activity, and it is acquired using the SCORM RTE. The Sequencing Definition describes how an activity must be sequenced, and is characterized by a set of rules in the SCORM manifest.

10.2.3 Learning Objects

After presenting the SCORM model, it is important to give at least a short overview of what can be done with it. The main outcomes of using the SCORM standards to develop educational content are learning objects. A LO can be described as a set of connected items with the purpose of providing instruction, sometimes practice and also assessment of educational objectives (Wiley 2000). There are two concepts in SCORM that are similar to the concept of LO. SCO and Assets are considered as LOs that are built using the SCORM standards.

Instead of the traditional several hour pieces of instruction, learning objects provide self contained, small and reusable units of instruction. LO can be grouped together in larger units to cover the curriculum of a course, and the material to be used. They are a new type of computer-based instruction inspired in the object-oriented paradigm (Wiley 2000) because object-orientation values the creation of components that can be reused in multiple contexts (Dahl and Nygaard 1966).

The key idea behind LO is that instructional designers can build small components of instruction (compared to the size of a complete course) that can be combined, reorganized and grouped in different ways, and therefore they raise issues of portability and interoperability. That is why the topic of learning objects is usually related to E-Learning standards.

Moreover, LOs are also understood as electronic units' deliverable over the internet where a large number of students can use them simultaneously, contrarily to conventional educational media that can only exist in one place at a time.

10.2.4 Intelligent Tutoring Systems

ITS are educational systems that can provide direct personalized instruction and feedback to students. In general, tutoring systems combine artificial intelligence techniques, cognitive psychology and also approaches from the learning sciences.

At least three classes of ITS are worth discussing: Completely dynamic simulation-based ITS, were student's action can fundamentally change the problem. Media-based adaptive interactive multimedia instruction, in which media-based content objects are dynamically stitched together to develop mastery of knowledge outcomes, and static coaching systems like the example provided in which students are asked to perform a skill in a controlled environment (see also the Carnegie Mellon Geometry and Algebra tutors). Indeed, ITS have been proved to be beneficial for learning in many domains such as physics (Vanlehn et al. 2005), programming languages (Corbett and Anderson 1992), and middle school math (Ritter et al. 2007). Traditionally, ITS are described as having four main modules (Brusilovsky 1994) (see Fig. 10.1), but in terms of functionalities, the two most important features are the inner loop, and the outer loop (Vanlehn 2006).

Essentially, the inner loop is responsible for giving appropriate feedback and hints when a student is working on an activity (generally using artificial intelligence). The inner loop can also assess the student's competence and register it on the student model, which is a repository of information about the student. The student model can be used by an ITS for taking pedagogical and also educational decisions. Additionally, all the information acquired by the inner loop can be used by the outer loop for task selection.

The main duty of the outer loop is to wisely select a task/activity for the student to work. The most important design issues are selecting a task intelligently and obtaining a set of tasks to select from. The outer loop deals with tasks, and the inner loop deals with steps within a task. The outer loop is executed once per task, while the inner loop is executed once per step. Theoretically, an ITS comprises two loops as illustrated in the following pseudo-code (Vanlehn 2006)

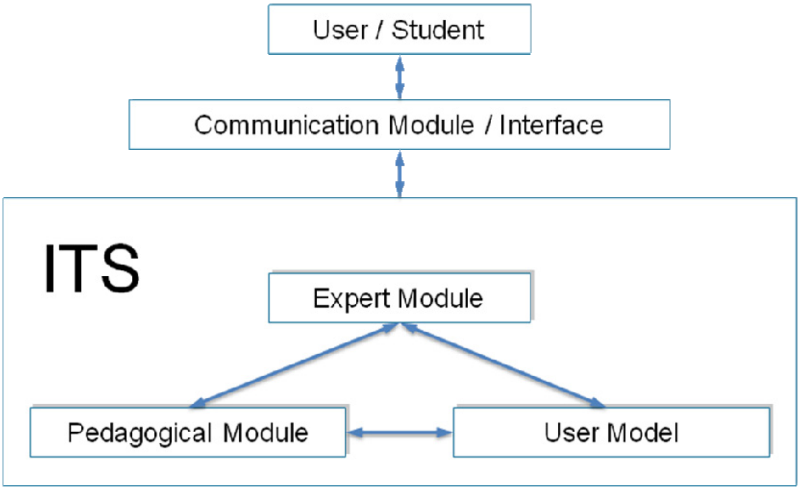


Fig. 10.1 Classic architecture of an ITS (Santos and Figueira 2011)

```
repeat until tutoring is done{
  select a task;
  repeat until task is done{
        tutor may provide a hint;
        student performs a step;
        tutor provides feedback about the step;
  }
  student submits his solution for the task;
}
```

A step is a user interface action that is part of completing a task/activity (e.g. solving a problem). Given that tutoring systems can support learning in many ways, when students are working on steps, let us simplify by assuming that ITS offer students services as they work. Some of the most common services (Vanlehn 2006) are:

- Minimal feedback on a step indicating only whether the step is correct or incorrect,
- Error-specific feedback on an incorrect step to help the student understand why a particular error is wrong,
- Hints on the next step to avoid students getting stuck on a step,
- Assessment of knowledge to evaluate answers that students give to steps,
- Review of the solution to help the student to understand the final solution of a problem.

The terms Inner Loop and Outer Loop should be understood as behavioural descriptions, rather than software structures. The software structures of an ITS can vary from one tutor to another and sometimes be very complex (Devedzic and Harrer 2005). However, the structure is generally irrelevant to the end user if the behavioural descriptions are implemented. One interesting software structure that is commonly used by researchers of the Pittsburgh Science of Learning Centre (PSLC, a joint research group between the University of Pittsburgh, Carnegie Mellon University and others) is to build tutoring artefacts that contain both graphical widgets, as well as everything that the tutoring system knows about that artefact such as correct steps, incorrect steps, hint sequences for various conditions, error-types and error-specific feedback (Vanlehn 2006). That is, a tutoring artefact represents a miniature tutoring system for a particular problem that has everything it needs to run (in one way, similar to a SCORM learning object).

Some educational systems lack an outer loop. For example, many courses that we see currently on e-learning platforms provide a table of contents where students can freely navigate. In these systems the content is previously selected by the instructor and is delivered online. Other systems can provide a very simple form of task selection (like one of the mechanisms found in Moodle), where the instructor defines the curriculum of a course week by week, and every week the respective content will become available in the platform. Some systems that do not lack an outer loop may, on the other hand, lack of an inner loop.

Systems that do not have an inner loop, but have an outer loop can assign tasks to students and collect the student's solution. Once the solution is submitted, these systems may either give the student another chance to solve the problem again (if the answer is wrong) or assign the student a new task. However, the lack of inner loop constructs such as intelligent step based problem solving support, hints and error specific feedback is evident.

Systems that lack an inner loop and can do task selection are generally called Computer-Aided Instruction (CAI), Computer-Based Training (CBT) or Web-Based Homework (WBH). Systems that do have an inner loop and can do task selection are called Intelligent Tutoring Systems. All the information in the next subsections of this chapter describing the outer loop is applicable to all the systems that can do task selection. The comments about the inner loop are only applicable to ITS.

10.3 Intelligent and Adaptive Attributes of the Approach

As mentioned before, the main objective of this chapter is to present the attributes of an approach that uses E-Learning standards to implement intelligent and adaptive educational systems that are interoperable and can be loaded into different platforms. The intelligent and adaptive educational systems that we are building with our approach are ITS, and therefore we focus on the implementation of the key features of ITS (*inner loops* and *outer loops*) to design learning objects.

The LO built with our approach, comply with SCORM standards and can be therefore accessed over the web, used in different platforms, and have their open source SCORM code reused for further development. It is important to state here that we employ a user modelling approach based on modelling skills. This method is very renowned, has been very successful for improving learning outcomes, and it is mainly used in a special type of ITS called Cognitive Tutors (CT) (Koedinger and Corbett 2006). Our approach is also based on a very well known paradigm for implementing ITS (Vanlehn 2006). This paradigm describes tutors in terms of required functionalities (or in other words, behaviour).

Our method for implementing tutors in terms of behaviour using the SCORM E-Learning standards is very distinct from previous approaches that tried to map ITS modules to SCORM modules (Sabbir Ahmed 2004), and extend the SCORM standards to support external user models or services (Rey-López et al. 2009, De Bra et al. 2010).

In our approach there is no mapping between the SCORM modules to ITS modules. In fact, in our approach there are no *modules*, but rather *models* (Santos and Figueira 2011). We also do not require the extension of the standards to rely on external databases for storing the user model. This is a key benefit of our approach, because SCORM allows using external resources to implement LO. However, this is strongly not recommended because using external resources significantly limits the portability of the educational content (Software 2011).

10.3.1 Approach Description - Inner Loops

First of all, the inner loop is responsible for providing problem solving support. This is usually done by delivering services to students, such as error specific feedback, hints, and assessment of knowledge. These services can be implemented in several ways, but in general they are rule based.

That does not mean that they must be implemented using a rule based language. In fact, many tutors have inner loops that are not implemented using rule based languages (Aleven et al. 2009). To design the inner loop services with our approach we recommend coding the tutors with ECMAScript language (since ECMAScript complies with the norms of the SCORM standards). To design the inner loop services we also strongly recommend running a Cognitive Task Analysis (CTA).

10.3.1.1 Cognitive Task Analysis

CTA consists of a set of techniques for describing knowledge, and also a set of strategies for knowledge engineering (Schraagen et al. 2000). Since CTA can be used for knowledge representation, it can be therefore used for the development of ITS (Lovett 1998) (after all, they are knowledge based systems). Using CTA is very popular for the development of ITS (Psotka et al. 1988). There are many examples of tutors designed with CTA such as (Koedinger and Corbett 2006, Corbett and Anderson 1992).

Basically, a cognitive task analysis can provide a set of guidelines for the implementation of the inner loops services, and it also gives a theoretical foundation to design the user interface, write the rules for hints, and error specific feedback (Santos and Figueira 2011).

In short, when it comes to performing a CTA it is important to stress that it involves identifying the components of a task that are necessary for adequate performance. CTA is therefore an important step for ITS design because it provides a decomposition of the curriculum, and also the identification of the skills that students should learn. Identifying the skills for the given domain is very important since they represent the heart of the student model (Vanlehn 2006).

In many situations the student model is a repository of data where tutors are able to record the skill grades that are involved in the domain of the ITS. Therefore, to represent our student model we estimate and store the proficiency of students in different skills. This is a very common user modeling approach that is used in many ITS, and became very popular in a specific type of ITS called CT (Anderson et al. 1995). CT are ITS built using CTA and the Adaptive Control of Thought (ACT) cognitive architecture (Anderson 1993). Some CT implement an adaptive task selection strategy called macro adaptation (Corbett and Anderson 1994) and to achieve this they use the "grades" of the domain skills to select tasks.

It is really important to stress here that to implement our student model and "record grades" we use the SN tracking data, more specifically we rely on SCORM objectives. Objectives are part of the SCORM tracking data and allow tracking the status of individual learning, and share this status across activities.

Objectives are able to store numerical values that represent grades (or in our case, proficiency of skills). While students are trying to solve problems in the SCOs/activities, the proficiency of the skills is being recorded in the SCORM objectives (using the SCORM ECMAScript functions of the RTE). To summarize, for implementing the inner loop we should first carry out a CTA, then implement all the inner loop functionalities with SCOs using ECMAScript and the SCORM RTE functions that fill in the student model using SN objectives (Santos and Figueira 2011).

10.3.2 Approach Description - Outer Loops

To implement the outer loop, in addition to SCORM objectives for representing the user model, we need the SCORM sequencing definition (SD). SCORM SD is a set of rules to describe how an activity should be sequenced. These rules can consult the information stored in the SCORM objectives to make decisions.

The fundamental mechanism for implementing outer loops is: 1) to state a set of sequencing rules; 2) use these rules to access the student model in run-time and read the values stored in the SCORM objectives; 3) based on the values that are recorded in the user model define what activity should be launched using the rules. If we compare the mechanism described here with the SCORM sequencing loop, it is easy to see that they are very similar. In fact, this is all that we can expect from an ITS outer loop, which is task selection.

No matter how simple this may look like, implementing an outer loop is definitely not an easy task. First of all, it requires knowledge about the set of activities available, and the ways how to alternate appropriately between them. Second, it is necessary to handle the innards of the user model for determining the most suitable activity for a particular user.

10.3.3 Approach Description - Prototype Demonstration

To clarify the way the intelligent and adaptive attributes of our approach work, let's take as an example a prototype that we built. First of all, our SCORM learning object (that is an ITS in this case) needs to be uploaded to any other SCORM compliant platform and made accessible to the target students. When a student logs in, and the ITS is loaded, what appears on the Web-Browser is illustrated in Fig. 10.2.

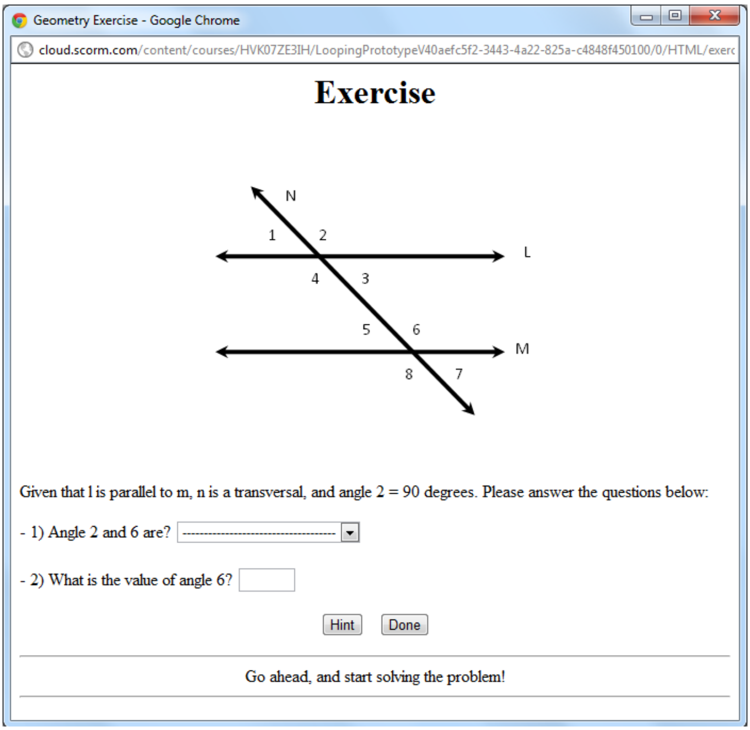


Fig. 10.2 Initial problem loaded in the LMS

Since it is the first time the student is loading the ITS, his user model is empty and our prototype cannot do adaptive task selection yet. Therefore the ITS selects the first problem that is available in the pool of existing exercises. As it is shown in Fig. 10.2 the domain of our prototype is angles formed by parallel lines. The skills that we are keeping track of in terms of the user model are:

- Identify corresponding angles,
- Calculate corresponding angles,
- Identify supplementary angles,
- Calculate supplementary angles,

- Identify vertical angles,
- Calculate vertical angles,
- Identify alternate exterior angles,
- Calculate alternate exterior angles,
- Identify alternate interior angles,
- Calculate alternate interior angles.

In this prototype our method for "grading" the skills in the user model is relatively simple, but it fits the purpose of demonstrating how our approach actually works. Skills are graded in a range of 0 to 100. Every step necessary to solve a problem has a corresponding skill in the user model.

Skills are all initialized on 20, and when a student gets a step correctly, the value of the corresponding skill is increased by 10 points. If a student gets a step wrong, the value of the corresponding skill is decreased by 5 points.

While students are working on the activities, the proficiency of the skills is being recorded/updated in the user model using the SCORM objectives with the SCORM functions of the RTE. For example, Fig. 10.3 shows the status of the ITS after a student has performed a step.

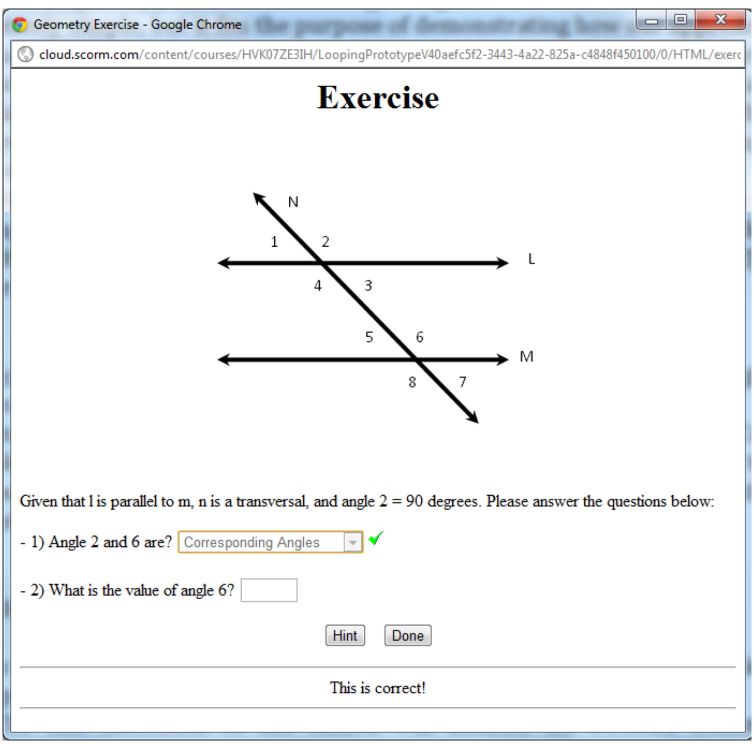


Fig. 10.3 ITS Status after the performance of a step

In the case presented in Fig. 10.3, the student has performed correctly a step about the skill "Identify corresponding angles", which has increased the value of this skill in the user model by 10 points. Let's say that the student continues working on this exercise, but he is having difficulty in solving the other step and therefore he asks for a hint.

In Fig. 10.4 we can see the ITS status after a hint request. The ITS can provide three levels of hints for each step. At each level hints get more detailed and the last hint is a "bottom out hint" (a hint that tells the student the solution for the step). When a "bottom out hint" is given, the student gets a penalty of 5 points (which is similar to answer a question incorrectly).

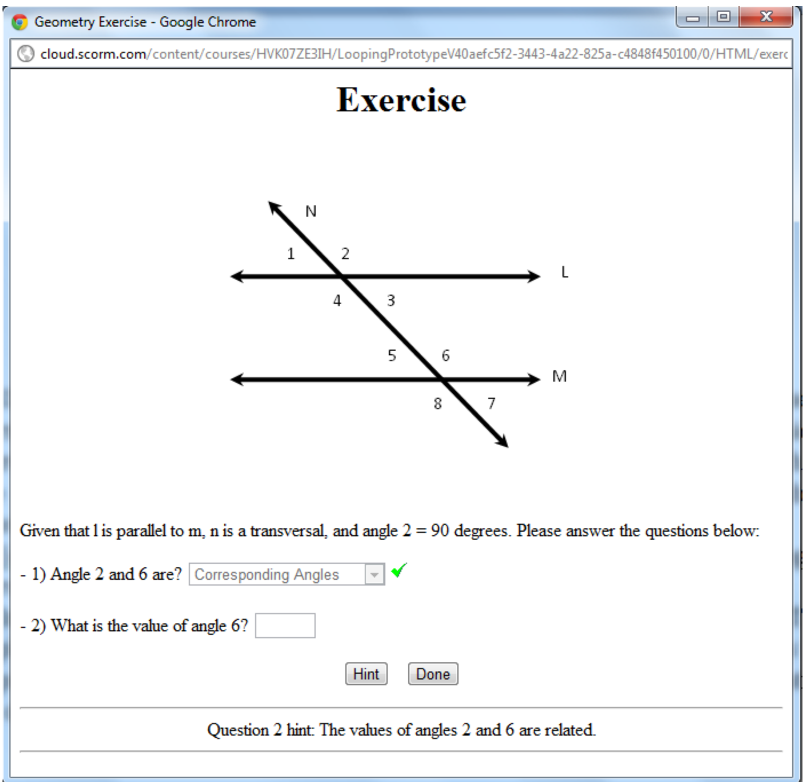


Fig. 10.4 ITS Status after a hint request

As mentioned before, the content of the hints of our ITS is determined using CTA and Artificial Intelligence (AI) techniques (such as knowledge engineering and rule based systems).

In fact, all the feedback given by the ITS is determined using AI rules. There are therefore rules to determine, when steps are performed correctly or incorrectly, rules to determine when hints should be given, rules to determine error specific feedback, etc. The rule based engine of our prototype is relatively flexible and allows the dynamic creation of exercises in run-time. This is very useful because the ITS can generate an infinite number of exercises using just a few templates.

To dynamically create exercises we need enough templates to cover the practice of all skills. When the ITS outer loop's rules use the user model to choose which skills should be practiced, a template is selected. When the template is loading in the Web-Browser through AJAX (Asynchronous JavaScript and XML) a random image is selected from a pool to represent the problem. Based on the selected image, pairs of angles are also randomly selected from a pool. Next, a random value is assigned to an angle, and then all the other angles can be calculated using AI rules. Let's see the intelligent and adaptive attributes of the approach by examining a situation where a student answers a question incorrectly.

Fig.10.5 shows the ITS status after a student gives an incorrect answer to a step. As we can see, this first error message does not give any details about the error. This is called on ITS terminology "minimal feedback". If the student gives another incorrect answer for this step, then the ITS will give a detailed description about the error.

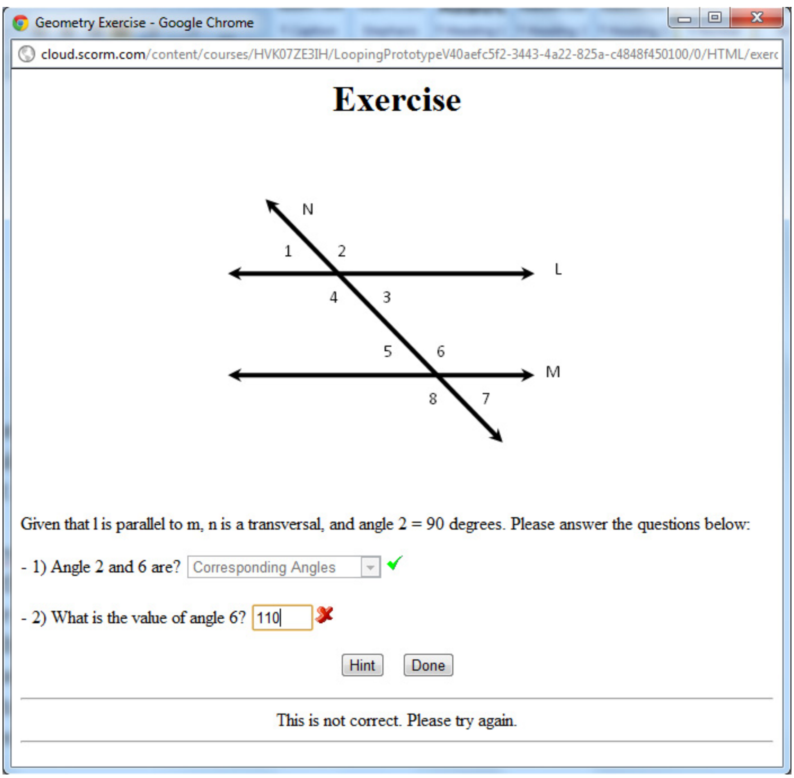


Fig. 10.5 ITS Status after a student error

Fig. 10.6 shows the ITS status after a student gives an incorrect answer to a step for the second time. This time the error message has specific details about the error, which in ITS terminology is called "error specific feedback". We have to stress that every time a student performs a step, the user model is always updated and the proficiency level of the skills changes. As mentioned before, the set of skill levels constitute our user model and therefore they will influence the task selection process.

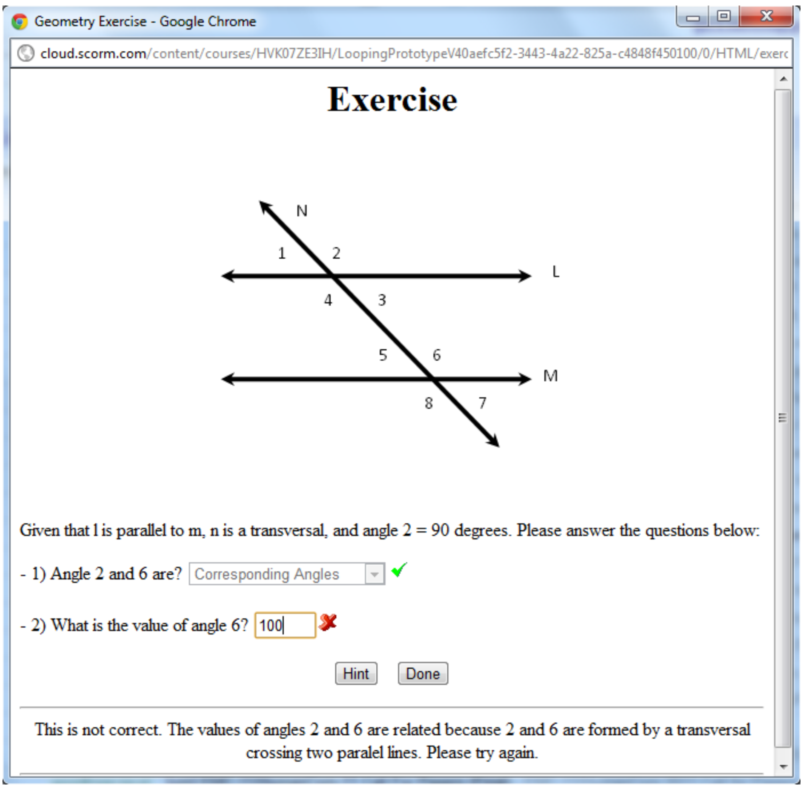


Fig. 10.6 ITS error specific feedback

To perform adaptive task selection, we basically have a set of rules that can check the values of the skills in the user model. Some of the rules are part of the SCORM sequencing rule set, but many task selection rules are manually written by the ITS knowledge engineer/instructional designer (see next section for more details). These rules determine which activity is more suitable for the student, and deliver it to the Web-Browser. This task selection process continues until all the domain skills have been mastered. When a student masters all the domain skills the outer loop terminates and the instructional process is completed.

10.3.4 Approach Description – Technical Details

This section briefly describes the innards of our prototype. The idea is to give at least a glimpse of how it actually works internally. Some technical details will be illustrated with code snippets from the system. For instance, let's consider the interface that was presented in the previous sub-Sect. in Fig. 10.2.

Every time a student works on a step, a JavaScript function is called to evaluate the user action. Every clickable widget in the user interface is actually associated with a function that will assess the action performed, and instantly give some feedback. The lines of code below show the HTML code related to *question 1 (that was presented in the last section).* As we can see, the JavaScript event *onChange* is associated to the function *question_1_rules*:

```
- 1) <span id="q1_wording"></span>
  <select id="question_1" onchange="question_1_rules()">
       <option selected>----------------</option>
       <option>Alternate Interior Angles</option>
       <option>Alternate Exterior Angles</option>
       <option>Supplementary Angles</option>
       <option>Vertical Angles</option>
       <option>Same Side Interior</option>
       <option>Corresponding Angles</option>
  </select>
  <span id="q1_img"></span>
<br/>
```

This excerpt of code above is associated to a template that as mentioned in the previous section can generate many different problems for the same set of skills. As we can see, each HTML element is associated to an ID and the respective content is determined dynamically allowing an infinite number of problems to be randomly generated. If a student tries to solve *question 1*, the function *question_1_rules* will be called and the respective JavaScript code will be executed. One of the first things that the function does is to execute the following line of code:

```
var indexOfIdentifyObjective = findObjective("Identify
Corresponding Angles");
```

The function *findObjective* will return a reference to a SCORM objective and the JavaScript variable *indexOfIdentifyObjective* will be used to manipulate the grades of the skill *Identify Corresponding Angles*. To obtain the level of proficiency of this skill (which is part of the user model) we have to execute the following code:

```
grade = parseFloat(doGetValue("cmi.objectives." + index-
OfIdentifyObjective + ".score.scaled"));
```

The variable *grade* holds now the value that is stored in the field *score.scaled* of the SCORM objective that corresponds to the skill *Identify Corresponding Angles*. The program can now manipulate this value and when it wants to update it in the user model, the following code has to be executed:

```
doSetValue("cmi.objectives." + indexOfIdentifyObjective +
".score.scaled", grade.toString());
```

When the skill is considered mastered by the ITS the following function is called to make that clear in the SCORM user model:

```
setObjToPassed(indexOfIdentifyObjective);
```

To perform task selection, every time an activity is terminated we have to call a LO that runs a task selection script. The script will use AI rules to determine the next suitable activity. For that, the script has to access all the values representing the proficiency level of skills, and therefore the beginning of the script is something like:

```
…
var indexOfIdentifyObjective1 = findObjective("Identify
Corresponding Angles");

grade1 = parseFloat(doGetValue("cmi.objectives." + index-
OfIdentifyObjective1 + ".score.scaled"));

var indexOfIdentifyObjective2 = findObjective("Calculate
Corresponding Angles");

grade2 = parseFloat(doGetValue("cmi.objectives." + index-
OfIdentifyObjective2 + ".score.scaled"));
…
```

After loading the proficiency of the skills to the local variables, a rule based engine will determine the next suitable activity. The set of rules for task selection have a structure that is similar to the following pattern:

```
…
if ( (grade1 < X && grade2 < Y) || (grade1 < Z) ||
(grade2 < W) ){
  selectActivity("activity1");
} else if ( (grade3 < X && grade4 < Y) || (grade3 < Z) ||
(grade4 < W) ){
  selectActivity("activity2");
}
```

10.3.5 Architecture and Software Structures

One of the most interesting aspects of our approach is that it suggests a new paradigm for the implementation of ITS (Santos and Figueira 2011). In terms of structures, ITS developed with our approach differ a lot from the classical ITS architecture (please consult Fig. 10.1 again). As we have mentioned before in this chapter, in our approach we do not have modules, but models.

This represents a major change in the way we structure our code in order to implement ITS. First of all, modules are separated software entities that communicated to each other using an API or communication protocol. Modules can be easily identified in the source code once they are implemented separately. On the other hand, models can represent human intentions, semantics or explicit formalizations required for solving specific problems.

Models are sometimes not so easily identified in the source code because they do not necessarily need to be implemented separately, and they do not need to use a communication protocol or API to communicate to each other. The models used to implement ITS in our approach are a result of running a CTA that provides us guidelines for: 1) the recommended ways to teach a specific domain – Pedagogical Model; 2) the recommended methods for solving problems in the domain – Expert Model; 3) the necessary skills involved in the domain – Student Model. Fig. 10.7 (Santos and Figueira 2011) pictures the architecture of a LO object implemented with our approach.

As we can see above, the boundaries between the Pedagogical Model and the Expert Model are not so distinct. This is because both models are coded together in the same LO that actually contains everything it needs to run (except for the Student Model which is consulted/updated in run-time using the SCORM RTE functions). Based on this concept of packing everything in one single object, we have introduced the idea of something that we call a Tutoring Artifact (TA) (Santos and Figueira 2011). A TA is nothing more than a LO with intelligent tutoring capabilities. Therefore, it contains small pieces of instruction.

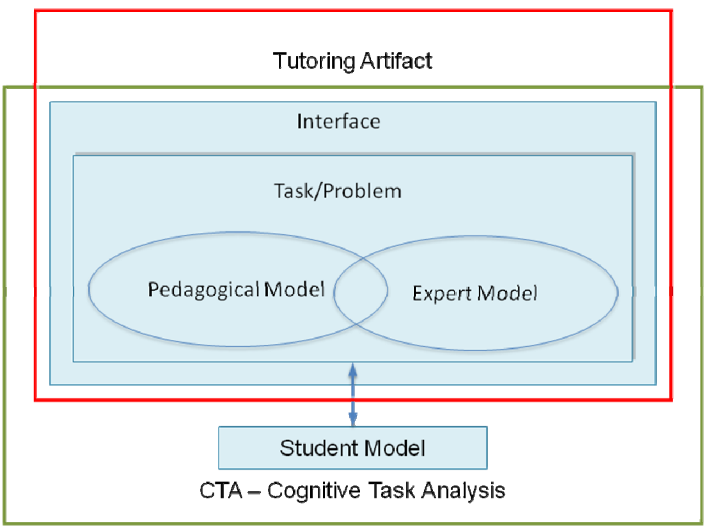


Fig. 10.7 Architecture of a LO/tutoring artifact (Santos and Figueira 2011)

As we have mentioned before in a previous chapter: "Our ITS architecture consists in building tutoring artifacts that contain graphical widgets, and also everything that a tutoring system knows about the artifacts such as, correct steps, incorrect steps, hint sequences, error-types and error-specific feedback Like a mini tutor. That is, a tutoring artifact represents a miniature tutoring system for a particular problem that has everything it needs to run. This approach is commonly used by researchers of the Pittsburgh Science of Learning Center, and it was has been very successful because it applies a divide and conquer strategy for implementing the tutors. By dividing a large tutoring system in several small tutoring artifacts, which can be grouped together to create an ITS as whole, it is possible to reduce the complexity of the development process (Santos and Figueira 2011)."

10.4 Evaluating the Approach

To evaluate, if our approach really allows the interoperability of ITS we have tested our ITS in different SCORM compliant educational platforms, different browsers, and different Operating Systems (OS). As expected, our prototype runs correctly in all SCORM compliant educational platforms, OS and Web browsers used for evaluation (Santos and Figueira 2011).

As we can see, our prototype is fully conformant to all three SCORM specifications. As a demonstration of interoperability, Fig. 10.8 shows one of the SCO loaded in the Odijoo E-Learning Platform using the Firefox Browser in a Windows 7 OS. In addition, to assure that our prototype is fully compliant to SCORM, we have performed a SCORM compliance test that is shown in Fig. 10.9. Fig. 10.10 shows one of the SCOs loaded in the SCORM SRTE using the internet Explorer Browser in a Windows XP operating system, and Fig. 10.11 shows our ITS loaded in the Rustici SCORM Cloud educational platform using the Google Chrome Browser in a Mac OS 10.5.8.

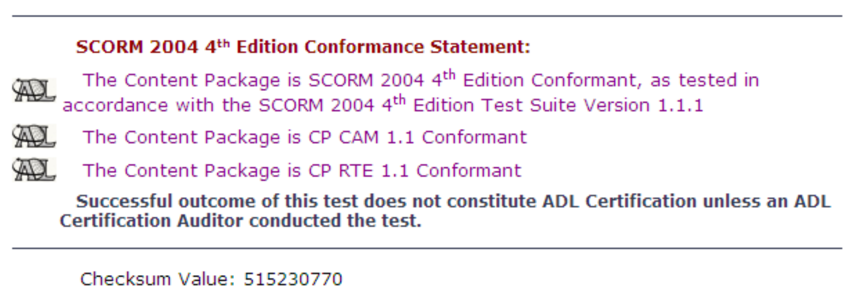


Fig. 10.8 SCORM Test Suit Conformance Test

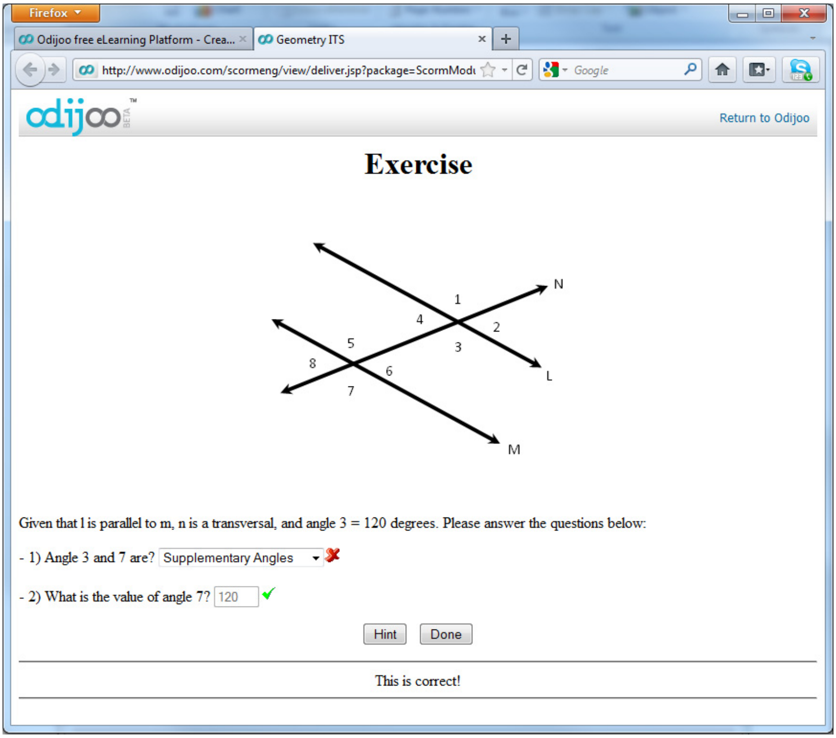


Fig. 10.9 ITS loaded in Odijoo E-Learning platform

To guarantee that all the features of the ITS work appropriately we have tested hints, error-specific feedback, correct and incorrect types of input, updates in the user model, task selection and dynamical creation of problems. Independently of the settings used, all features of the ITS work as expected.

10.5 Discussion

As the majority of the educational systems currently available mainly contain plain educational content (like PDF and PPT), it is desirable that ITS could be loaded in these systems. However, ITS have interoperability issues and cannot be uploaded to most of the educational platforms (Rey-López et al. 2009).

One of the ways to solve the interoperability issues of ITS is by using E-Learning standards, which are in general responsible for guaranteeing interoperability and reusability of educational content. However, supporting ITS is not the focus of current E-Learning standards (Rey-López et al. 2009).

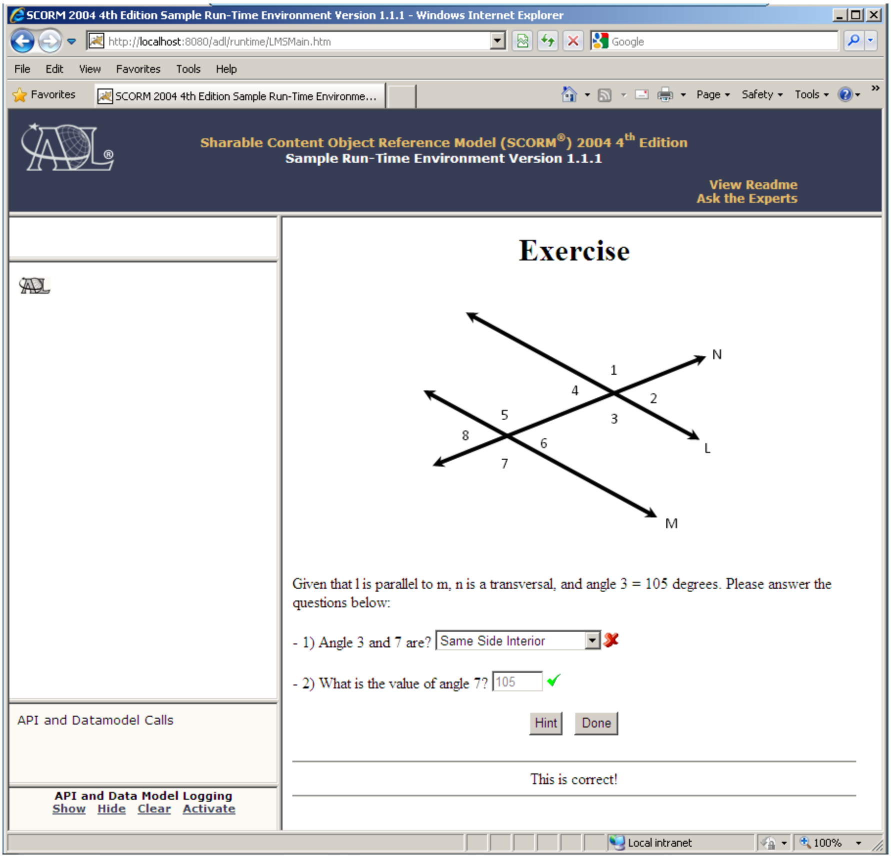


Fig. 10.10 ITS loaded in the SCORM SRTE

Despite the fact that ITS are outside of the scope of E-Learning standards, it is possible to implement ITS using them. SCORM 2004, which is also known as SCORM 1.3, introduced a complex idea named sequencing and navigation, and based on the sequencing specification we have developed an approach for implementing ITS (Santos and Figueira 2010a, Santos and Figueira 2011).

The approach developed by the authors differs from the previous attempts to implement ITS with standards, because it is not based on mapping ITS modules to SCORM modules (Santos and Figueira 2010), neither on extending the SCORM standards (M. Rey et al. 2006, De Bra et al. 2010) to support detailed profile management using external databases/systems (which dramatically effects the portability of the educational content) (Software 2011).

We use constructs such as SCORM objectives to model the students and store what we need to know. The user modeling approach is based on supporting cognitive models that record the proficiency levels of skills in the student model.

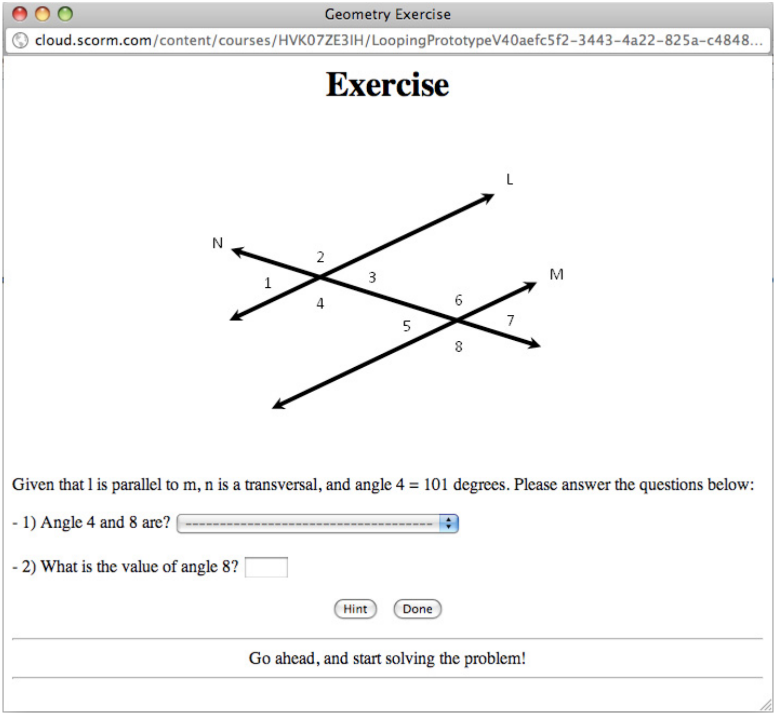


Fig. 10.11 ITS loaded in the Rustici SCORM cloud educational platform

This user modeling technique is well established (Corbett and Anderson 1994, Corbett et al. 2000), has been studied for several years (Koedinger and Corbett 2006), and proved to be very efficient when it comes to improving learning outcomes (Ritter et al. 2007, Vanlehn et al. 2005, Corbett and Anderson 1992). In addition, the approach is based on supporting the behavioral features of ITS (Vanlehn 2006). This paradigm based on functionalities, is a famous model for building tutors, and since its publication it has dominated the top conferences on the ITS field.

In short, our approach innovates the development of ITS by designing a novel approach for building tutors. The approach is unique and distinguishing because: 1) it uses de-facto E-Learning standards without any modification or extension to implement ITS; 2) it is grounded on a sound student modeling technique; 3) it is grounded on a renowned paradigm for the implementation of ITS. In addition, we have designed a new paradigm in terms of ITS architectures. Our architectural structure is composed of models and not modules. These models are used to implement self contained intelligent LO that we call Tutoring Artifacts. These TA are mini tutoring systems that can be grouped together to create an entire course (like a normal LO).

We have to stress that our approach is not designed to support all kinds of adaptive and personalized educational systems. We are definitely not presenting a generic solution that can be used for the implementation of many different adaptive educational systems. In fact, we are presenting an approach that allows the implementation of ITS with pure de-facto E-Learning standards, using cognitive models, and the ITS paradigm based on functionalities.

When ITS are built based on our approach, they are compliant with E-Learning standards and can be therefore be accessed over the web, used in different platforms and have their open source code shared by communities.

10.6 Conclusions and Future Work

This chapter presents an approach for implementing interoperable ITS using the SCORM e-learning standards and describes a prototype that was implemented using this approach. We highlight the intelligent and adaptive attributes of our approach, and we also explain how they work in practice.

Our approach shows that it is possible to use e-Learning standards (without any modification or extension) to implement intelligent and adaptive educational systems which are interoperable and can be deployed into different educational platforms. In short, our approach innovates on the development of ITS by designing a novel method for building tutors. The approach is unique and distinguishing because in addition to using E-Learning standards without any external resources or services, it is based in a sound student modelling technique (using CTA), and a renowned ITS paradigm (based on functionalities).

When ITS are built with the approach, they are compliant with E-Learning standards and can be therefore accessed over the web, used in different platforms, and have their open source code shared by communities.

At this stage we have only evaluated the interoperability aspects of our ITS (experimenting it in different educational platforms in a "laboratory setting"). As future work, we are preparing experiments to evaluate the effectiveness of the ITS in a real educational setting. Our goal is to have students on the domain using the platform so we can discuss the effectiveness of the ITS developed.

References

ADL. ADL - Who We Are, `http://www.adlnet.gov/About/Pages/Default.aspx` (acessed February 01, 2011)

ADL. SCORM, `http://www.adlnet.gov/Technologies/scorm` (acessed February 01, 2011)

ADL. SCORM Benefits, `http://www.adlnet.gov/Documents/SCORM%20FAQ.aspx#scormq2` (acessed February 01, 2011)

Aleven, V., McLaren, B.M., Sewall, J.: Scaling up programming by demonstration for intelligent tutoring systems development: An open-access website for middle-school mathematics learning. IEEE Transactions on Learning Technologies 2(2), 64–78 (2009)

Aleven, V., McLaren, B.M., Sewall, J., Koedinger, K.R.: A new paradigm for intelligent tutoring systems: Example-Tracing Tutors. International Journal of Artificial Intelligence in Education 19(2), 105–154 (2009)

Anderson, J.R.: Rules of the mind. Erlbaum, Hillsdale (1993)

Anderson, J.R., Corbett, A.T., Koedinger, K.R., Pelletier, R.: Cognitive tutors: Lessons learned. The Journal of the Learning Sciences 4(2), 167–207 (1995)

Beatty, B., Ulasewicz, C.: Faculty perspectives on moving from blackboard to the Moodle learning management system. TechTrends: Linking Research & Practice to Improve Learning 50(4), 36–45 (2006)

Brusilovsky, P.: The construction and application of student models in intelligent tutoring systems. Journal of Computer and Systems Sciences International 32(1), 70–89 (1994)

Brusilovsky, P., Wade, V., Conlan, O. (2007). From learning objects to adaptive content services for E- Learning. In: Architecture Solutions for E-Learning Systems. IGI Global, USA (2007)

Corbett, A.T., Anderson, J.R.: The LISP intelligent tutoring system: Research in skill acquisition. In: Larkin, J., Chabay, R., Scheftic, C. (eds.) Computer Assisted Instruction and Intelligent Tutoring Systems: Establishing Communication and Collaboration. Erlbaum, Hillsdale (1992)

Corbett, A.T., Anderson, J.R.: Knowledge tracing: Modeling the acquisition of procedural knowledge. User Modeling and User-Adapted Interaction 4, 253–278 (1994)

Corbett, A.T., McLaughlin, M., Scarpinatto, K.C.: Modeling student knowledge: cognitive tutors in High School and College. In: Proceedings of UMUAI, pp. 81–108 (2000)

Dahl, O.-J., Nygaard, K.: SIMULA: an ALGOL-based simulation language. Commun. ACM 9, 671–678 (1966)

De Bra, P., Smits, D., van der Sluijs, K., Cristea, A., Hendrix, M.: GRAPPLE: Personalization and Adaptation in Learning Management Systems. In: Proceedings of WCEMHT, pp. 3029–3038 (2010)

Devedzic, V., Harrer, A.: Software patterns in ITS architectures. International Journal of Artificial Intelligence in Education 15(2), 63–94 (2005)

Downes, S.: E-learning 2.0. In: Proceedings of e-Learn. (2005)

Hrastinski, S.: Asynchronous & Synchronous E-Learning. EDUCAUSE 31(4), 51–55 (2008)

Koedinger, K.R., Corbett, A.T.: Cognitive tutors: Technology bringing learning science to the classroom. In: Sawyer, R.K. (ed.) The Cambridge Handbook of the Learning Sciences, pp. 61–78. Cambridge University Press (2006)

Lovett, M.C.: Cognitive Task Analysis in Service of Intelligent Tutoring System Design: A Case Study in Statistics. In: Goettl, B.P., Halff, H.M., Redfield, C.L., Shute, V.J. (eds.) ITS 1998. LNCS, vol. 1452, pp. 234–243. Springer, Heidelberg (1998)

Rey, M.L., Fernanndez, A., Diaz, V.R., Pazos, J.A.: Providing SCORM with adaptivity. In: Proceedings of ICWWW, pp. 981–982 (2006)

McCarthy, J.E.: Military Applications of adaptive training technology. In: Lytras, D.G., Ordóñez de Pablos, P., Huang, W. (eds.) Technology Enhanced Learning: Best Practices, pp. 304–347. IGI Global (2008)

Mia, S.: The Difficulties in web-based tutoring, and some possible solutions. In: Proceedings of the Workshop of IESWWW (1997)

Psotka, J., Massey, L., Mutter, S.: Intelligent tutoring systems: lessons learned. Lawrence Erlbaum Associates, Hillsdale (1988)

Rey-López, M., Brusilovsky, P., Meccawy, M., Díaz-Redondo, R., Fernández-Vilas, A., Ashman, H.: Resolving the problem of intelligent learning content in learning management systems. International Journal on E-Learning 7(3), 363–381 (2008)

Rey-López, M., Díaz-Redondo, R.P., Fernández-Vilas, A., Pazos-Arias, J.J., García-Duque, J., Gil-Solla, A., Ramos, C.M.: An extension to the ADL SCORM standard to support adaptivity: The t-learning case-study. Computer Standards & Interfaces 31(2), 309–318 (2009)

Ritter, S., Anderson, J.R., Koedinger, K.R., Corbett, A.: Cognitive tutor: Applied research in mathematics education. American Psychonomic Bulletin & Review 14(2), 249–255 (2007)

Sabbir-Ahmed, K.: A conceptual framework for web-based intelligent learning environments using SCORM-2004. In: Proceedings of IEEE ICALT, pp. 12–15 (2004)

Santos, G., Figueira, A.: Reusable and inter-operable web-based intelligent tutoring systems using SCORM 2004. In: Proceedings of ECe-L, pp. 521–528 (2010a)

Santos, G., Figueira, A.: Web-based intelligent tutoring systems using the SCORM 2004 specification: A conceptual framework foriImplementing SCORM compliant intelligent web-based learning environments. In: Proceedings of ICALT, pp. 676–678 (2010b)

Santos, G., Figueira, A.: Intelligent tutoring systems with SCORM. The European Journal for the Informatics Professional 7(2), 34–42 (2011)

Schraagen, J., Chipman, S., Shalin, V., Shalin, V.: Cognitive task analysis. L. Erlbaum Associates (2000)

Software, R. SCORM Content packaging (2011), http://scorm.com/scorm-explained/technical-scorm/content-packaging/ (acessed November 26, 2010)

van Harmelen, M.: Design trajectories: four experiments in PLE implementation. Interactive Learning Environments 16(1), 35–46 (2008)

Vanlehn, K.: The Behavior of tutoring systems. International Journal of Artificial Intelligence on Education 16(3), 227–265 (2006)

Vanlehn, K., Lynch, C., Schulze, K., Shapiro, J., Shelby, R., Taylor, L., Treacy, D., Weinstein, A., Wintersgill, M.: The Andes physics tutoring system: five years of evaluations. In: Proceedings of ICAIE, pp. 678–685 (2005)

Watkins, R.: E-Learning. Handbook of improving performance in the workplace: selecting and implementing performance interventions, pp. 577–597. John Wiley & Sons, Inc. (2010)

White, S.: Higher education and learning technologies an organisational perspective. ECS, University of Southampton (2006)

Wijekumar, K., Meyer, B., Felix, D., Walker, D.: Creating web-based intelligent tutoring systems using the. net infrastructure: A case study. In: Proceedings of WCE-LCGHHE, pp. 2494–2497 (2003)

Wiley, D. (ed.): The instructional use of learning objects. Association for Educational Communications and Technology (2000)

Abbreviations

ACT	Adaptive Control of Thought
ADL	Advanced Distributed Learning
AI	Artificial Intelligence
AJAX	Asynchronous JavaScript and XML
API	Application Programming Interface
CAI	Computer-Aided Instruction

CAM	Content Aggregation Model
CBT	Computer-Based Training
CT	Cognitive Tutor
CTA	Cognitive Task Analysis
ITS	Intelligent Tutoring system
LMS	Learning Management System
LO	Learning Object
OS	Operating System
PC	Personal Computer
PLE	Personal Learning Environment
PSLC	Pittsburgh Science of Learning Center
RTE	Run-Time Environment
SCO	Sharable Content Object
SCORM	Sharable Content Object Reference Model
SD	Sequencing Definition
SN	Sequencing and Navigation
TA	Tutoring Artifact
WBH	Web-Based Homework
XML	Extensible Markup Language

Part III
Virtuality

Chapter 11
Real Classrooms in Virtual Worlds: Scaffolding Interdisciplinary Collaborative Writing

Reneta D. Lansiquot

New York City College of Technology of The City University of New York
300 Jay Street, Brooklyn, NY 11201, USA
rlansiquot@citytech.cuny.edu

Abstract. This chapter advocates the use of a mixed methodology approach to classroom-based research while highlighting a related, three-year study that used virtual worlds to facilitate writing. Second Life (SL), a three-dimensional virtual world created entirely by its residents, afforded students the ability to quickly and collaboratively model scientific processes. The inherent social nature of this virtual space served as an effective mode of fostering collaborative writing, scaffolding academic research, and lowering writing apprehension in interdisciplinary technical writing courses. Additionally, the adaptive attributes of this virtual world reveal the role of student avatars as intelligent agents who can use model-centered instruction and design layering in this learning system.

11.1 Introduction

Effective educational models must provide multiple representations of content, avoid oversimplifying the content, and support context-dependent knowledge. To this end, knowledge sources should be highly interconnected. Constructed knowledge is important; however, students must have an opportunity to develop their own representations, or models, of information (Anderson and Krathwohl 2000).

Compared to two-dimensional representations, three-dimensional, intelligent, and adaptive virtual spaces provide a vivid and realistic environment and facilitate the creation of a correct and complete mental model of collaborative tasks. The malleability of avatar appearance and identity with the freedom to customize that appearance can create an unbiased cultural experience. In technical writing courses, students communicated technical and scientific information to audiences through written and oral presentations using multimedia simulations.

For their final project, students explored SL[1], an online three-dimensional virtual world created entirely by its residents, individuals in the real world. This application was chosen over open-source three-dimensional modeling tools such as Blender[2] because of the social interaction and interactivity that is possible in this

[1] www.secondlife.com
[2] blender.org

A. Peña-Ayala (Ed.): Intelligent and Adaptive ELS, SIST 17, pp. 269–292.
springerlink.com

virtual world. SL also has the added advantage of allowing students to create three-dimensional models without needing to learn the often over-burdening complexities of industrial-strength applications straightaway. This is particularly important because the course is primarily a writing course, which is housed in an English department. As SL is composed of islands owned by not just individuals but also academic institutions, game companies, researchers, and the like, users (student residents) are able to visit islands and interact with inhabitants. Students can teleport to locations related to their researched scientific process description and discuss their topic with residents, such as designers and scientists to gather information and gain multiple perspectives.

Following the discussion, and using an image as a guide, students created a three-dimensional model of their scientific process in SL and developed a guided multimedia simulation of this process. As they developed their model with the group, students noted the ways in which the process worked and created an instructional manual for their peers. In this way, instructional manuals were created by student groups for other students. The manual included an explanation of how their scientific process worked and how to recreate their three-dimensional model. Later, student group manuals incorporated online multimedia elements.

11.1.1 Model-Centered Instruction

A set of principles to guide instructional designers in selecting and arranging design constructs, thus appropriately called a *design* theory. Model-Centered Instruction (MCI) favors designs that originate with and maintain the priority of models as the central design structure.

Providing a layered view of design, MCI assumes that a designer organizes constructs within several, somewhat independent layers characteristic of instructional designs: the model/content layer, the strategy layer, the control layer, the message layer, the representation layer, the media-logic layer, and the management layer. The designer selects and organizes structures within each layer in the process of forming a design. The designer also aligns the structures within layers with those of other layers to create a vertical modularity in the design that improves its manufacturability, maintainability, and reusability of designed elements. Various characteristics typify a design layer, such as: characteristic design goals, building block constructs, design processes, design expression and construction tools, and principles to guide the arrangement of structures.

Over time, a layer becomes associated with skill sets, publications, and a design culture. Instructional theories provide principles to guide design within one or more of these layers, but no theory provides guidelines for all of them, suggesting the wisdom of subscribing to multiple local theories of design rather than a single monolithic theory.

The description of Model-Centered Instruction, as any design theory, can be in terms of the prescriptive principles it expresses for each of the following layers.

- Content: The content of instruction should be perceived in terms of models of three types: 1) models of environments; 2) models of cause-effect systems (natural or manufactured); and 3) models of human performance. Together, these constitute the elements necessary for performance and, therefore, for learning. The expression of content should be relative to the full model structure rather than simply as facts, topics, or lists of tasks,
- Strategy: The strategy of instruction should be in terms of problems. The definition of a problem is any self-posed or instructor/designer-posed task or set of tasks formed into structures called work models (Gibbons et al., 1995). These are essentially scoped performances within the environment, acting on systems and exhibiting expert performance. Problems may be worked examples or examples for the learner to work. During problem solution, the learner may request or the instructor may offer instructional augmentations of several kinds. Dynamic adjustment of work model scope is an important strategic variable,
- Control: Control (initiative) assignment should represent a balance between learner and instructor/designer initiatives calculated to maximize learner momentum, engagement, efficient guidance, and learner self-direction and self-evaluation. Instructional controls (manipulative) should allow the learner maximum ability to interact with the model and the instructional strategy's management,
- Message: Contributions to the message arise from multiple sources, which may be architecturally modularized: 1) from the workings of the model; 2) from the instructional strategy; 3) from the controls management; 4) from external informational resources; and 5) from tools supplied to support problem-solving. The merging of these into a coherent, organized and synchronized message requires some kind of message or display management function,
- Representation: MCI makes no limiting assumptions about the representation of the message. Especially with respect to model representation, it anticipates a broad spectrum of possibilities—from externalized simulation models to verbal "snapshots" and other symbolics that call up and use models learners already possess in memory,
- Medial-Logic: MCI makes no assumptions regarding the use of media. Its goal is to achieve expressions that are transportable across media. The selection of the model and the problem as central design constructs assist in this goal,
- Management: MCI makes no assumption about the data recorded and used to drive instructional strategy except to the extent that it must parallel the model's expression of the content and align with the chosen units of instructional strategy.

The purpose of this chapter is to illustrate how mixed methodology is the ideal approach to classroom-based research. This methodology is especially important in interdisciplinary courses. What is more, the use of virtual worlds is a useful way to facilitate communication and creation. It supports bringing theories such as cognitive flexibility theory into practice.

This theory, as well as the benefits of interdisciplinarity, is discussed bellow. A case is then made for the use of mixed methodology for classroom-based research. Virtual worlds are then added into the mix because the use of avatars as intelligent agents who can create objects support students majoring in science, technology, engineering, and mathematics in technical writing classes as these students come to the classroom with high writing apprehension. Statistical results reveal significantly lowered student writing apprehension, and excerpts from a sample student group final instructional manual project are included to illustrate the shift from theory to practice, showing successful student collaborative writing.

11.2 Conceptual Framework

Interdisciplinarity focuses on questions, problems, or topics too complex or broad for a single discipline or field to cover adequately, and thrives on drawing connections between seemingly exclusive domains. Interdisciplinary courses improve student learning. As a result of interdisciplinary learning, students: 1) recognize varied perspectives; 2) purposefully connect and integrate across-discipline knowledge and skill to solve problems; 3) synthesize and transfer knowledge across disciplinary boundaries; 4) become flexible thinkers; 5) gain comfort with complexity and uncertainty; 6) comprehend factors inherent in complex problems; 7) become lifelong learners; 8) apply integrative thinking to problem-solving in ethically and socially responsible ways; 9) think critically, communicate effectively, and work collaboratively. Usually theme-based, interdisciplinary courses address a complex question or problem that requires engaging various disciplines (Lansiquot et al. 2011). Educators offer course materials in a meaningful way and promote the transfer of knowledge and student engagement through connections between different disciplines.

Educators need effective cognitive and pedagogical strategies to scaffold interdisciplinary studies. Situated learning contends that knowledge needs an authentic context for novices to become part of a *community of practice* and that learning requires interaction and collaboration (Lave and Wenger 1991). Vygotsky (1978/2006) viewed interaction with peers as an effective way of developing skills and strategies, and suggested that educators should employ cooperative learning activities for less competent learners to develop with help from more skilful peers within the Zone of Proximal Development (ZPD). He posited the ZPD as the area where the most sensitive instruction or guidance should be, allowing the learners to develop skills to use on their own to develop higher mental functions.

In a constructivism theoretical framework, learning is an active process in which learners construct new ideas or concepts based on their current and past knowledge. The learner selects and transforms information, constructs hypotheses, and makes decisions, relying on a cognitive structure to do so. Cognitive structure (i.e., schema, mental models) provides meaning and organization to experiences and allows the individual to "go beyond the information given" (Bruner 1973).

Therefore, educators should encourage students to discover principles by themselves while engaging in active dialogue. The task of the instructor is to translate information into a format appropriate to the learners' current state of understanding. In other words, instructors should organize the curriculum in a spiral in order for students continually to build on what they have already learned. According to

Bruner (1966), instruction must be: 1) concerned with the experiences and contexts that make the student willing and able to learn (readiness); 2) structured so that it can easily be grasped by the student (spiral organization); 3) designed to facilitate extrapolation and or fill in the gaps (going beyond the information given). Bruner (1986, 1990, 1996) later expanded this theoretical framework to include social and cultural factors.

Cognitive flexibility theory builds on constructivist theory. It stresses the importance of constructed knowledge; learners must develop their own representation of information in order to learn properly. The theory focuses on the nature of learning in complex and ill-structured domains. Spiro and Jehng (1990) stated: "By cognitive flexibility, we mean the ability to spontaneously restructure one's knowledge, in many ways, in adaptive response to radically changing situational demands. This is a function of both the way knowledge is represented (e.g., along multiple rather single conceptual dimensions) and the processes that operate on those mental representations (e.g., processes of schema assembly rather than intact schema retrieval)" (p. 165).

The theory is largely concerned with transfer of knowledge and skills beyond their initial learning situation. For this reason, emphasis is on the presentation of information from multiple perspectives, including a variety of case studies that present diverse examples. The theory also asserts that effective learning is context dependent, therefore instruction needs to be very specific.

The design of cognitive flexibility theory supports the use of interactive technology. Its core principles include: 1) learning activities must provide multiple representations of content; 2) instructional materials should avoid oversimplifying the content domain and support context-dependent knowledge; 3) instruction should be case-based and emphasize knowledge construction, not transmission of information; and 4) knowledge sources should interconnect, rather than viewed as compartmentalized.

11.3 Mixed Methodology in Action

When a researcher enters a complex and interdependent classroom environment hoping to explore the effectiveness of a problem-solving education technology program, that researcher must carefully consider the appropriate types of research methods. In the current study, the need to investigate learning as it occurred in real-world situations led to a classroom-based intervention, which Brown (1992) termed *design experiments*. According to Brown: "Classroom life is synergistic: Aspects of it that are often treated independently, such as teacher training, curriculum selection, testing, and so forth actually form part of a systemic whole."

Just as, it is impossible to change one aspect of the system without creating perturbations in others, so too it is difficult to study any one aspect independently from the whole operating system. Thus, we are responsible for simultaneous changes in the system, concerning the role of students and teachers, the type of curriculum, the pace of technology, and so forth. These are all seen as inputs into the working whole.

Although a design experiment may connote experimental methods because it takes a problem-solving approach to identify possible solutions, it is less formal than the proposal of Campbell and Stanley (1963) for *true* experimental designs, including both pre- and post- tests, or utilizing the former or the latter, and controls for validity. In addition, a true experimental design "is more iterative, dependent on a series of hypothesis probes over time" (Cook et al. 2003, p. 34). Consider the case of a researcher who enters a classroom to evaluate an educational learning system and share perspectives on three separate occasions.

The foremost concern of a researcher, as an evaluator, is construct validity, including varying sub-types nested under the broad term, which accumulates evidence that the assessment behaves in expected ways (Weiss 1998). Researchers use different research methods from purely quantitative to purely qualitative or something between in the classroom, including randomized experiments, case studies, or an ethnography.

A researcher may enter a classroom with the belief that randomized experiments should be more frequent in education as well as in work on educational learning systems (Cook et al. 2003). Some critics may argue that randomized experiments involve withholding possible beneficial treatments from people who need them. On the other hand, an advocate of randomization may point out that the purpose of the experiment is to discern the effectiveness of treatments. Federal policy for gathering evidence of *what works* in education includes emphasizing randomized field trials as the preferred method to generate scientific evidence on the effectiveness of educational programs (Chatterji 2004). Unfortunately, this kind of research design promotes a one-day, one-visit study that may overlook specific proclivities of diverse classroom groups.

A second researcher may enter the classroom intent on capitalizing on the strengths of the case study method, that is, its ability to examine a case in-depth, within its *real-life* context (Yin 2006).

Case study methodology focuses on similarities and differences within the observation. The similarity principle states that one can discover the meaning of a symbol by analyzing its similarity to other symbols. The contrast principle states that one can discover the meaning of a symbol by analyzing its difference from other symbols (Tashakkori and Teddlie 1998). Thus, ambiguity and complexity become the order of business. Interdisciplinary efforts plus diversity interrupts the ways of looking at the world. In order to interrupt our current way of thinking, we need a shift toward interdisciplinary research. In collecting case study data and in all qualitative research, the researcher must *triangulate* or establish converging lines of evidence to make findings as robust as possible (Yin 2006).

A third researcher, an ethnographer, may enter the classroom with the Chatterji (2004) notion of triangulation, which establishes construct validity through convergence in findings. Thus, configurational validity is: "A theory of making meaning in a digital world—a world in which different authors use images stored in digital video to layer and reconstitute original documentation". Configurations are combinations or arrangements of disparate views that can be assembled and reassembled in any number of patterns […] the views of multiple authors can be layered in clusters or constellations so that larger, more representative theories may begin to unfold. (Goldman-Segall 1998) point out: “When different constellations are gathered, layered, and analyzed new patterns emerge” (p. 261).

Ethnography is good for generating hypotheses and allows ongoing analyses, but it may not test hypotheses (Anderson-Levitt 2006). It depends primarily on two broad methods, participant observation and open-ended interviewing. Analysis of this kind relates to narrative inquiry, the study of experience as story. Consequently, it is foremost a way of thinking about experience. Narrative inquiry as a methodology entails a view of phenomena. To use narrative inquiry methodology is to adopt a particular view of experience as phenomena under study (Connelly and Clandinin 2006). According to Connelly and Clandinin, capturing the profound differences in kinds of narrative inquiry is in the distinction between living and telling. In narrative inquiry, four terms (e.g., living, telling, retelling, and reliving) structure the process of self-narration.

To bridge the gap between design experiments and ethnographic narrative accounts, Goldman (2004) suggested an approach called design ethnography, which did not use an experimental method as the foundation, but rather used observation, participation, recording, description, interpretation, and conclusions. The goal is the creation of thick, rich descriptions of events (Geertz 1973). According to (Cobb et al. 2003): “Design experiments, on the other hand, develop theories; but not merely empirically tune *what works*”. However, design experiments are not experiments in the technical sense in which statisticians use the term (Cook et al. 2003). Design experiments can play several important roles within a program of research that can include random assignment experiments and “can be viewed as valuable precursors to randomized experiments but should not be regarded as alternatives to them” (Cook et al. 2003, p. 35). Fortunately, research camps agree on the length of time needed. That is, extended-term mixed methods (ETMM) and design ethnography call for a three-month study immersion or a semester.

Unfortunately, researchers cannot infer varied attributes, motivations, and feelings from behaviors. Observations are more useful if they combine with other methods of collecting information (Tashakkori and Teddlie 1998). An important question for researchers becomes what are the implications, not merely the observations, of the research.

The identity of the aforementioned three researchers and their chosen research methods are the same—the whole researcher has come to the classroom. (McCandliss et al. 2003) reported: “The design experiment–laboratory science collaboration process perhaps best conceived as a dialogue in search of common ground, rather than as a unidirectional transfer of information from ‘research-to-application’ or a request for a lab to test a specific hypothesis” (p. 15). Quantitative tradition—the positivists and post-positivists—and qualitative tradition—the constructivists—can exist in a symbiotic relationship. It is not about choosing one type of research over another. Generalizing from case studies, for instance, reflects substantive topics or issues of interest, and offers logical inferences (Yin 2006), while ethnography combines with case study methodology.

The primary objective of alternative analytic strategies is to enable the researcher to use both types of analysis simultaneously or in sequence in the same study. Mixed investigations simultaneously use both types of data collection (qualitative and quantitative), and both types of data analysis, statistical and thematic analysis (Tashakkori and Teddlie 1998). In this way, mixed methodology is not an additive, with alternative research approaches included in a problem-based study, but is, instead, part of the initial groundwork in the beginning. In other words, the intent is not to *save* a study in which current findings show no quantitative significant differences. For classroom-based research, using mixed methods is an ideal approach.

11.4 Adding Virtual Reality to the Mix

The integration of virtual worlds provides new spaces for discussions and overcomes geographic limitations. In virtual worlds, students see all avatars interaction is more explicit, and students use gestures rather than relying purely on text-based interactions. They do not need to rely on conceptual ideas of presence, such as aliases in a knowledge forum (Padmanabhan 2008). This technology encourages students to transfer what they learn in a virtual world to traditional academics.

In technical writing courses, which are inherently interdisciplinary, merging communication and the science, technology, engineering, and mathematics disciplines, students communicate technical and scientific information to their chosen audiences through written and oral presentations. Students focus on questions, problems, or topics too complex or broad for a single discipline, and can, therefore, thrive on drawing connections between seemingly exclusive domains.

Using multimedia, simulations, and industry-standard software, students also analyze science and technology articles, model technical communication, and practice collaborative research, writing, and presentations. For their manual, group work, and projects, students had the next instructions:

1. With your group mates and on your own, explore SL and teleport to locations related to your scientific topic. Discuss your topic with SL residents to gather information, narrow your topic, and gain user perspectives. To what kind of audience are you catering, technical or non-technical?
2. Create a three-dimensional model of your scientific process in SL, and develop a guided simulation. Use Linden Scripting Language (LSL) to incorporate interactivity (e.g., moving objects, communicating with avatars, or both). While you are developing your model with your group, take notes on each step,
3. Write a manual[3] for your peers on your topic, including how your scientific model works and how to create it in SL. The manual should contain information and instructions (how-to, tips) that are useful for and understandable by your chosen student audience. Include appropriate graphics, such as snapshots taken of the process. Your preface should specify your audience and purpose.

---

[3] Students conducted usability testing and revised their manual accordingly. To provide further guidance to their target audience, student groups developed online multimedia versions of their instructional manuals (e.g., websites, instructional videos, etc.) including their interactive 3D model. Students adhered to instructional design principles to reduced cognitive load.

By the end of the semester, students in the interdisciplinary laboratory course should be able to:

- Communicate clearly in technical writing and in oral presentations,
- Learn, use, and explain technical documents,
- Develop instructional manuals,
- Create guided scientific process simulations,
- Research information independently,
- Write collaborative technical documents,
- Use professional tools for technical communications.

11.5 Learning Outcomes

The first year of the research focused on examining writing apprehension among junior and senior undergraduate students from five sections of the course. Most of these students were computer programmers, enrolled in majors for which high writing apprehension is particularly acute. The data were collected through a writing apprehension test (Daly and Miller 1975, Reed et al. 1988). This yearlong study—"A Student's Guide to Virtual Worlds"—revealed a significant reduction in student writing apprehension.

Results from the first semester were statistically significant, $t(34) = 4.120$, $p < .000$. Overall, for the first year of the study, when the focus was on writing apprehension, the decrease was also statistically significant, $t(59) = 4.306$, $p < .000$. The subsequent two years of this three-year study focused on scaffolding collaborative writing in these interdisciplinary courses through the continued use of creating three-dimensional models, simulations of scientific processes in the virtual world SL. By placing themselves in the role of author, students addressed writing apprehension by directing their prose not to a teacher or other evaluator, but to other students who would look to them for guidance.

11.5.1 Collaborative Writing

Throughout the semester, assignments were scaffolded to help students complete the final project, an instructional manual. For example, students evaluated technical documents based on detailed criteria; they also created original objects and wrote a procedural document for building their object. Halfway into the semester, students were first charged with researching, writing, and presenting a scientific process description report on a topic of their choice. On their own, they were to briefly describe a scientific process and a problem associated with this process.

Students were required to include evidentiary support that defines the nature of the problem and, based on the evidence, determines a possible solution to the problem. Students modeled this process by choosing a representative illustration, and explaining this image during a pitch presentation to their classmates. Finally, students formed groups of about three based on their interest in one of the selected instruction-manual scientific process topics.

Each member of the group conducted further research on this scientific process by finding, evaluating, reading, and summarizing at least three relevant sources and then submitting an annotated bibliography of his or her research. Furthermore, students were required to post all files in the group discussion area online.

In this way, during the initial research for the project, it was clear which students were contributing and which students were not. In addition, each member of the group included a cover memo describing the group writing process for this assignment, and his or her individual contributions.

As a group, students wrote their table of contents, that is, an outline for their instruction manual project, which allowed them to distribute tasks. Each member of the group was then responsible for writing an agreed-upon section of the manual. Additionally, students assigned roles such as editor, proofreader, and so on.

Students prefaced their instructional manuals with directed notations on how to best use the information that followed. Below in subsequent sections leading up to the discussion, a sample excerpts from a student group manual is given.

11.5.1.1 Key Questions for the Definition of the Manual

In this section a set of questions are made in order to guide the definition of the manual as follows:

- Who should use this manual? This manual is intended for a wide range of users, from the novice to the intermediate. We do not recommend this manual to an expert or to a user who is adept in SL, because we will be going through it through the eyes of a beginner and will not be covering any advanced aspects of SL,
- What product, procedure, or system does the manual describe? The manual will first describe how hydroelectricity works, and subsequently describe the mechanisms, functions, and features of the three-dimensional model of the hydroelectric power plant,
- What is the manual's purpose? The manual's purpose is to guide the user through the construction of a hydroelectric power plant simulation,
- What are the manual's major components? The manual's major components are simply a description of the subject (hydroelectric power plant), the construction of its actual model in the virtual space, steps to adding functionality to model through coding, and lastly the addition of any finishing touches,
- How should the manual be used? The manual should be used as both a step-by-step guide to construct the three-dimensional model as well as an information source for any future SL projects a user might have. We do not recommend using it as a manual to learn SL itself, as it is not a manual for teaching SL. Therefore, even though the manual will be very clear and concise, as well as provide information on how to get the user started in SL, it is recommended that the user spend a brief amount of time tinkering with, and testing out, the program itself. This is especially important since the step-by-step guide is heavy with coordinate instructions.

11.5.1.2 Introduction to Hydroelectricity and the Problems in Nepal

Imagine waking up one morning without any electricity. You had soon notice how heavily compromising this would be. Residents in Kathmandu Valley woke up to this exact situation one Wednesday morning, as the Nepal Electricity Authority (NEA) began its 51 hour weekly power outage across the country. The demand for power in Nepal is estimated to be 845 (MW), while the NEA is only able to provide 450 MW. The electricity issue in Nepal has resulted in key problems, such as political instability, a lack of jobs, and economic turmoil.

The situation does not seem to be improving either, as Sher Singh Bhat, a senior official of the NEA, states: "The situation could ease a little bit in the summer, but the power cuts are here to stay for another five to six years". Nepal, which has several large and small rivers crisscrossing the country has the potential to generate close to 80,000 MW of power, but only a fraction of this has been exploited. The utilization of hydroelectric plants will not only provide much-needed electricity for Nepal, but also answer many of problems that Nepal currently faces.

Hydroelectricity is a renewable source of energy and the hydropower plants uses mechanics to harness the energy water creates and then to convert it into electricity. Hydroelectric plants are dependant on water flowing through a dam, which turns a turbine, and subsequently a generator. Most hydropower plants rely on a dam that holds back water, creating a large reservoir. Gates on the dam open and gravity pulls the water through the penstock, which is a pipeline that leads to the turbine. Water builds up pressure as it flows through this pipe. The water then strikes and turns the large blades of a turbine, which is attached to a generator. While the turbine blades are turning, a series of magnets inside follow suit, turning rapidly. Giant magnets rotate past copper coils, producing alternating current by moving electrons. The transformer inside the powerhouse takes the alternating current and converts it to a higher voltage current. Used water is run through pipelines and re-enters the river downstream. The water in the reservoir is considered potential energy, which is inactive energy. When the gates are opened, the water flowing through the penstock becomes kinetic energy due to motion. It is this kinetic energy that is converted to electricity.

Students included additional detailed information to this introductory section, then provided a model illustration of their scientific process. After the initial description, students focused on describing the process of designing a three-dimensional model of this image, which served as the center of their simulation, concentrating on the most basic steps first.

11.5.2 Building in Second Life

Primitives—or prims for short—are the building blocks for everything constructed in SL. Prims come in fifteen different shapes and can be colored and textured to resemble almost anything. You can build anywhere in SL where you have permission to build. The best place to learn to build in SL is its public build zone known as the *sandbox*, where it is necessary to take into account the next tips:

1. In the sandbox right-click and select *Build* of the drop down menu,
2. A *Build Toolbox* window (see Fig. 11.1) will open on screen and your mouse cursor will turn into a magic wand. Click on the ground with the wand to create a wood-textured prim cube (see Fig. 11.2),
3. To edit your prim, right-click on it and select *Edit* from the drop down menu. This will expose red, blue and green drag arrows used to move the prim. Red is the x-axis. Green is the y-axis. Blue is the z-axis,
4. You can edit a prim's size and position through the *Object* tab and edit the prim's textures in the Texture tab,
5. Click on a colored arrow to move the prim in that direction,
6. To change the prim's size, click on the *Object* tab in the *Build Toolbox* window and change the numbers in the X, Y, Z dimensions in the *Size* textboxes.

Now that the user is familiar with the construction of each discrete, individual shape, one can see that the whole model is made up of various compound objects and each compound object is made up of various prims (see Fig. 11.3).

11.5.2.1 Making Utility Poles with Power Lines

With regard to this project, the utility pole will be by far the simplest of the prims you will create, as it is shown in Fig. 11.4.

11.5.2.2 Making the Poles

In the constructions of poles eleven tasks are accomplished as follows:

1. In a clear area near the model of the hydropower plant right-click on the ground and select *Build*,
2. In the build toolbox window select the *Create* button and then select the *Object* tab,
3. Click on the cylinder icon and then left-click on an open area on *the grid* creating a prim of a cylinder object,
4. Right-click on your new prim and select *Edit* from the drop down menu,
5. Select the *Object* tab and under *Size* (meters) change the z – coordinate to read 10.0,
6. Select the *Texture* tab and click on the color picker (white box) and select the color brown from the palette, and then the click *OK* button,
7. To create the "cross" of the utility pole we will need to create another cylinder prim. Repeat steps 1-4,
8. Select the *Object* tab and under *Size* (meters) change the x– coordinate to read 0.20, the y – coordinate to read 0.20 and the z – coordinate to read 3.5,
9. Under *Rotation* (degrees) change the x– coordinate to read 150, the y – coordinate to read 90 and the z – coordinate to read 225,
10. Select the *Texture* tab and click on the color picker (white box) and select the color brown from the palette, and then the click the *OK* button.

With the prim selected and using the exposed red, blue and green drag arrows, move the prim around, align the horizontal prim near the top most part of the vertical prim until the middle of the horizontal prim is evenly "merged" into the top part of the vertical prim creating a "cross". It is convenient to leave enough space on top of the vertical prim as so it does not resemble the letter "T".

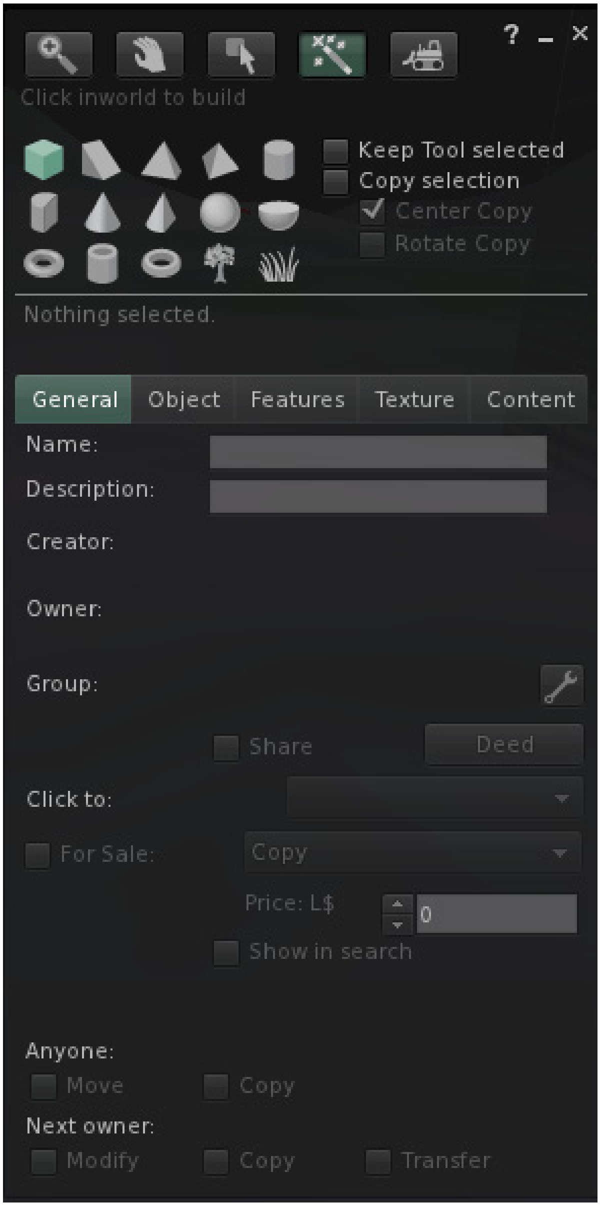


Fig. 11.1 Build Toolbox

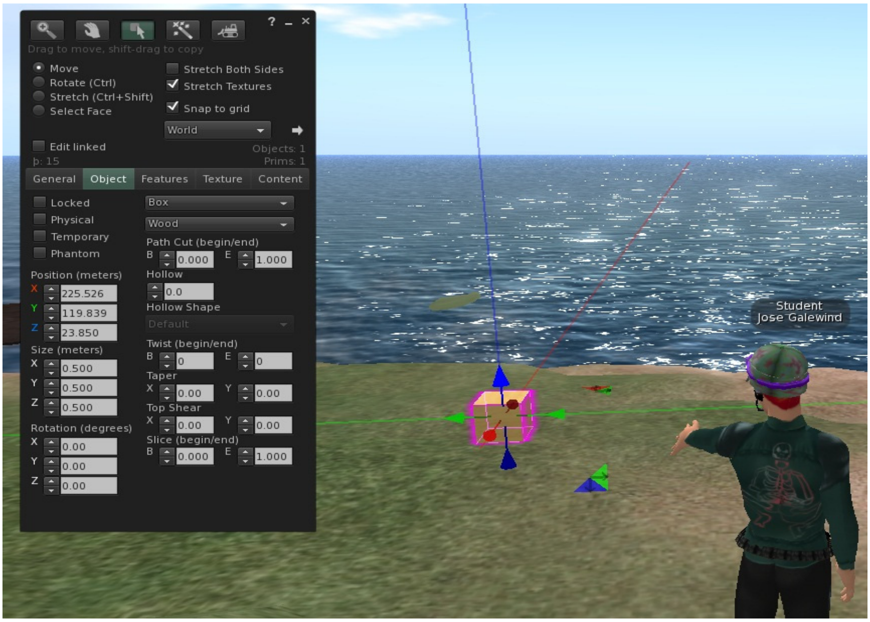


Fig. 11.2 Dropping first prim

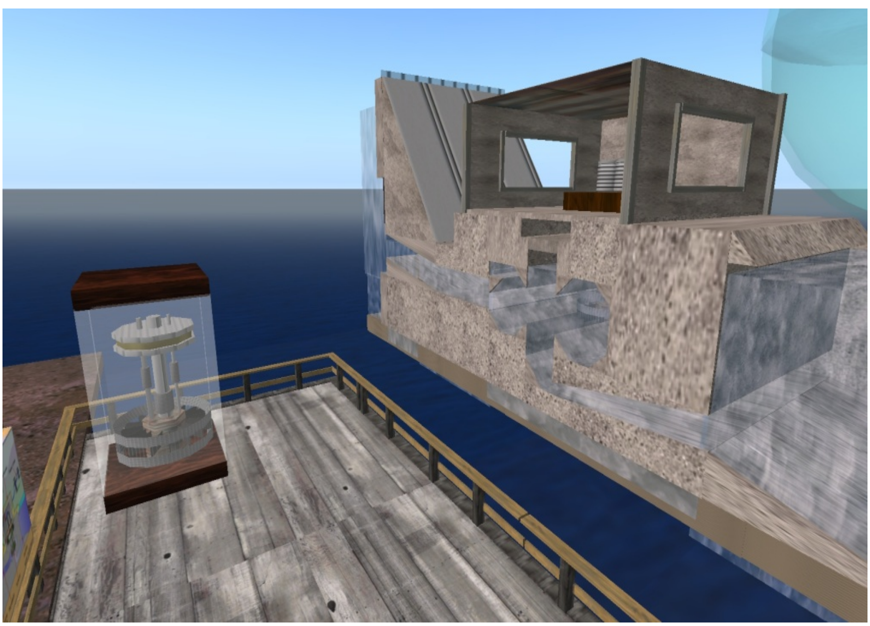


Fig. 11.3 Three-dimensional student group model of a scientific process, hydropower

Fig. 11.4 Complete utility poles with power lines

11.5.2.3 Making the Conduits

Concerning the design of conduits the next procedure is achieved:

1. Now to create the conduit to hold the power line, in a clear area near the model of the hydropower plant, right-click on the ground and select *Build*,
2. In the *Build Toolbox* window select the *Create* button and then select the *Object* tab,
3. Click on the cylinder icon and then left-click on an open area on "the grid" creating a prim of a cylinder object,
4. Right-click on your new prim and select *Edit* from the drop down menu,
5. Select the *Object* tab and under *Size* (meters) change the x– coordinate to read 0.15, the y – coordinate to read 0.15 and the z – coordinate to read 0.15,
6. Select the *Texture* tab and click on the texture picker (brown box) and in the *Pick: Texture* menu click on the *Filter Texture* textbox and type gray, then select *Gray Metal Scratched* in the solution box and click the *OK* button.

11.5.2.4 Duplicating a Prim

A couple of actions are made in order to duplicate a prim as it is shown next:

1. We need two conduit prims; therefore, we make a duplicate of the prim we have just created. This can be done by right-clicking on your conduit prim and select *Edit* from the drop down menu,

2. Holding down the *Shift* key and using the exposed red, blue or green drag arrows, move a duplicate copy of the prim away from the original.

11.5.2.5 Adding a Script

With the aim to add a script seven operations are carried out as follows:

1. To add a script to create a power line, from the *Build Toolbox* window of one of the conduit prims, select the *Content* tab,
2. Click on the *New Script* button to create a script called "New Script", then double click it to open up the script in a new window,
3. Select the default text within the script and press the *Delete* key on the keyboard to delete it,
4. Copy and paste the "Rope Constraint Script" by Comrade Podolsky into the current script and click on the *Running* and *Mono* checkboxes to remove the checks and click on the *Save* button, once the save is completed, click on the "*x*" icon on the upper right hand side of the window to close it,
5. Repeat steps 1-3 on the other conduit prim. This script needs to be altered slightly so as not to "connect" the power line to the other conduit on the same pole. This commented code is provided in Sect. 11.5.3,
6. With one of the conduit prims selected, right-click on it and select *Edit*. Now using the exposed red, blue and green drag arrows to move the prim around to attach it to one end of the horizontal pole on the main utility pole,
7. Apply step 6 to the other prim and link it to the horizontal pole's opposite end.

11.5.2.6 Linking Your Prims

The final procedure consists of linking the prims in the way explained in the next relation of actions:

1. Now that the utility pole has been created, all of the prims are linked together to make one unique object,
2. By right-clicking on the vertical horizontal pole and selecting *Edit* from the drop-down menu, hold down the *Ctrl* key and individually click on all of the other prims that make up the utility pole (the horizontal pole and each of the conduits). Afterwards all of the highlighted prims glow yellow,
3. Once they are highlighted, while holding down the *Ctrl* key, press the *L* key to link all of the prims together. When all of the prims are being linking, the linked object will glow blue; except the last item that was selected, this will glow yellow but still be linked,
4. Lastly, right-click on the utility pole and select *Edit* on the drop-down menu and using the exposed red, blue and green drag arrows, move the utility pole so it is clearly visible and completely above ground.

11.5.3 Linden Scripting Language

LSL is a language that gives behavior to SL objects and avatars. A component of this collaborative writing project in advanced technical writing classes required students to discuss their project with SL residents and to incorporate interactivity into their simulation.

Although this is not primarily a computer programming course, LSL facilitated student discussions to adapt available code for their project, in this case, attaching functioning simulated power lines to their hydroelectric power plant.

Students chose to create power lines that supported charges at their base. To accomplish this, students modified existing code to gain the necessary rope functionality for these power lines. As part of their manual, students appended the following commented code:

```
// Pseudo-Realistic Rope Constraint by Comrade Podolsky
// Free for non-commercial use.
// And of course, anyone is free and encouraged to
// modify this script for more specific applications
integer lockon = FALSE;

// The code creates the length of the simulated rope.
// It will stretch slightly longer than this, though.
float rope_length = 7.0;

// The following code dampens a fraction of the object's
// velocity every 0.1 seconds, if the rope is stretched
float dampening = 0.04;

// This code dictates how much the object "bounces" back
// after the rope is stretched to the limit of its
// length. This applies if the rope is stretched
// suddenly. Setting the floating bounce to 0.4 means it
// will bounce back with 40% of its original velocity.
// Setting this to below 0 will make the rope's act of
// constraint more flexible than it would normally be.
// Setting this to below 0 will make the rope's act of
// constraint more flexible than it would normally be.
float bouncing = 0.4;

// Here is the code to set the force constant of the
// rope. This applies when the rope is stretched slowly;
// in this case, it acts like a spring.
float constant = 16.0;
key target;
vector ropecolor;
list rope_effect = [];
```

```
default
{
    state_entry()
    {
        // This code will automatically locate "other"
objects. This is
        // important as students used the electricity
from the power lines
        // to charge simulated hybrid cars.
        llListen(123456,"","","TEST123");
        llSetTimerEvent(5.0);
    }
    timer() { if ( !lockon )
       {llWhisper(123456,"TEST123"); } }

    // Although this part of the code is rather resource
    // consuming, it does provide for a more real
    // simulation for students. For slower-moving
    // objects and for objects roped to something
    // static, range may be decreased and delay may be
    // increased. Sensor range must be at least twice
    // the length of the rope. Stretching the rope
    // farther will cause it to act like it has broken,
    // but it will restore itself if the objects are
    // moved back together. Therefore, students had to
    // account for the spacing of the poles and power
    // lines (i.e., the ropes).

    listen(integer chan, string name, key id,
           string msg)
    {
        if (!lockon)
        {
            lockon = TRUE;
            llWhisper(123456,"TEST123");
            llSetTimerEvent(0.0);
            target = id;
            llSensorRepeat("", target, SCRIPTED, 20.0,
                            PI, 0.1);
    // This is the particle effect representation of the
    // rope. Change the colors, alphas, scales, and
    // textures to the student choice. Here again, the
    // ability to code in this functionality allowed
    // students to provide more real simulations based
    // on their un-derstanding of how their chosen
    // scientific process works.
            rope_effect =
            [
                PSYS_PART_FLAGS,
                PSYS_PART_FOLLOW_SRC_MASK |
                PSYS_PART_FOLLOW_VELOCITY_MASK |
                PSYS_PART_TARGET_LINEAR_MASK |
```

```
                PSYS_PART_TARGET_POS_MASK,
                PSYS_SRC_PATTERN,
                PSYS_SRC_PATTERN_DROP,
                PSYS_SRC_TARGET_KEY,   target,
                PSYS_PART_START_COLOR, ropecolor,
                PSYS_PART_END_COLOR,   ropecolor,
                PSYS_PART_START_ALPHA, 1.0,
                PSYS_PART_END_ALPHA,   1.0,
                PSYS_PART_START_SCALE, <0.05,1.0,1.0>,
                PSYS_PART_END_SCALE,   <0.05,1.0,1.0>,
                PSYS_SRC_TEXTURE,      "",
                PSYS_PART_MAX_AGE,     0.5,
                PSYS_SRC_BURST_RATE,   0.001,
                PSYS_SRC_BURST_PART_COUNT, 1
            ];
            llParticleSystem(rope_effect);
        }
    }

    sensor(integer num_detected)
    {
        vector i_pos = llGetPos();
        vector u_pos = llDetectedPos(0);
        if (llVecMag(u_pos-i_pos)>rope_length)
        {
         // This code defines how an ideal spring would
         // behave under the noted conditions. Without
         // this statement, the rope would stretch
         // slowly, but indefinitely. As ropes do not
         // work this way, students are able to work
         // under real conditions and constraints.
         llSeForce(constant*llGetMass()*llVecNorm
           (u_pos-i_pos)*(llVecMag(u_pos-i_pos) -
            rope_length),FALSE);

         // This code dampens the motion of the object,
         // preventing buildup of excess kinetic energy
         // through oscillation.
            llApplyImpulse( llGetMass() * llGetVel() *
            dampening * -1.0 , FALSE );

         // Next, the direction of the object should be
         // controlled to prevent unwanted move-ment.
            vector wrongway = llVecNorm(i_pos - u_pos);

         // The velocity with which the object is
         // apparently moving, against the pull of the
         // rope, is another controlled variable.
            float wrongmag = ( llGetVel() -
                       llDetectedVel(0) ) * wrongway;
```

```
            // The following code stops the object from
            // stretching too much farther than rope
            // length.
               if ( wrongmag > 0.0 ) { llApplyImpulse
                    ( llGetMass() * ( ( -1.0 - bouncing ) *
                    ( wrongway * wrongmag ) ) , FALSE ); }
           }

            // If the two objects are closer than the rope
            // length, the rope does not do anything.

           else { llSetForce( ZERO_VECTOR, FALSE ); }
       }
       no_sensor()
       {
           llSetForce( ZERO_VECTOR, FALSE );
        // If the other object is nowhere to be seen, this
        // object exhibits no behavior,
       }
   }
```

For empirical evaluation, this study, which used a mixed-methodology approach and added virtual reality into the mix, included reviewing student instructional manuals that required collaborative writing, and building in SL and LSL. As the aforementioned section on learning outcomes detailed, writing apprehension was lowered for undergraduate students majoring in computer systems technology, and similar fields, in interdisciplinary advanced technical writing classes. Further significance of the outcomes assessed will be discussed below.

11.6 Discussion

During the first year of this research study, and the subsequent two years, student experiences in virtual communities allowed them to mirror academic argumentation through the following:

- Selecting their own topics, each group defined the scope of its topic to be manageable within the time allotted and considering the group members' expertise. Individual students presented their instructional manual topic proposal, based on the following instructions: Briefly describe a scientific process and a problem associated with this process. Make sure that you have evidentiary support to define the nature of the problem. Based on this evidence, what is a possible solution to this problem? Finally, model this process,
- Immersing themselves in the virtual learning communities under investigation and recording field notes of the experience,
- Interviewing experts, residents of the virtual world. These interviews, students pointed out, helped refine the topics for their manuals, and view objects in the virtual world as crafted art, not as technical artifacts created along the way to community building,

- Looking for areas of possible replication or gaps in knowledge, students realized the effective accomplishments in SL, including expertise on which they could build, and what was necessary to add,
- Creating a prototype and conducting usability testing, an iterative process that involved feedback, students experimented with their product, testing and refining it until they complete their manual. Manuals had to include clear replicable procedures. As the students conducted usability tests, they realized what information and tasks would help their users. Frustration and initial setbacks in performing a task in SL were recycled into explanations for new users, including tips and scaffolds not unlike the teaching implications educational researchers provide in their formal research studies. Conducting research to create a manual in SL helped students to experience the relevance of the traditional research structure (Lansiquot 2011, Schmid 2008),
- Presenting their findings to classmates,
- Concluding with a demonstration, discussion, and defense.

By the end of the course, students had completed the analysis, design, development, implementation, and evaluation phases used by instructional designers, student groups adhered to the Model-Centered Instruction.

The principles of model-centered instruction include:

- Experience: Learners should have maximum opportunities to interact for learning purposes with one or more systems or models of systems of three types: environment, system, and/or expert performance. The terms model and simulation are not synonymous; models can express a variety of computer-based and non-computer-based forms,
- Problem-solving: Interaction with systems or models should be focused by the solution of one or more carefully selected problems, expressed in terms of the model, with solutions determined by the learner, by a peer, or by an expert,
- Denaturing: Models denature from the real by the medium. Designers must select a level of denaturing matching the target learner's existing knowledge and goals,
- Sequence: Problems should be in a carefully constructed sequence for modeled solution or for active learner solution,
- Goal orientation: Problems should be appropriate for the attainment of specific instructional goals,
- Resourcing: The learner needs problem-solving information resources, materials, and tools within a solution environment (which may exist only in the learner's mind) commensurate with instructional goals and existing levels of knowledge,
- Instructional augmentation: The learner should have support during solutions in the form of dynamic, specialized, designed instructional augmentation.

The models created by the students were facilitated by SL via its three-dimensional modeling tool based around simple geometric shapes that allow avatars to build virtual objects. The most complex images, such as the model of a hydroelectric plant in Fig. 11.3, are all composed of incremental simple shapes (see Fig. 11.2 and Bonsor 2001).

11.7 Conclusions

In an interdisciplinary, advanced technical writing course, students had a space in which to create their own world and to improve their collaborative writing and academic research through a unique final group project. Project requirements incorporated their prior knowledge, but anticipated that students had little or no experience with SL prior to the start of the course. Thus, students had to work in groups to design their three-dimensional models using their avatars and to use their newfound knowledge to teach others by creating a detailed instructional manual and an interactive multimedia version.

This chapter discussed a three-year mixed methodology research study that focused on this final project, highlighting student construction of educational models in this virtual world that successfully supported their collaborative writing and academic research, while decreasing their writing apprehension. Additionally, students use of and need for model-centered instruction and design layering in a learning system was illustrated.

The ability to create and shape the environment and model aspects of the real world while going beyond the physical limitations in virtual space can provide an ideal scaffold for interdisciplinary courses. Although the technique may be useful in other disciplines, it shows particular promise for interdisciplinary studies, which involve two or more academic subjects or fields of study that synthesize broad perspectives, knowledge, skills, and epistemology in an educational setting.

References

Anderson, L.W., Krathwohl, D.R. (eds.): A taxonomy for learning, teaching and assessing: A revision of Bloom's taxonomy of educational objectives. Pearson, New York (2000)

Anderson-Levitt, K.M.: Ethnography. In: Green, J.L., Camilli, G., Elmore, P.B. (eds.) Handbook of Complementary Methods in Education Research, pp. 279–296. Lawrence Erlbaum Associates, Mahwah (2006)

Bonsor, K.: How hydropower plants work (2001), `http://science.howstuffworks.com/environmental/energy/hydropower-plant1.htm` (accessed December 19, 2011)

Brown, A.L.: Design experiments: Theoretical and methodological challenges in creating complex interventions in classroom settings. Journal of the Learning Sciences 2(2), 141–178 (1992)

Bruner, J.: The culture of education. Harvard University Press, Cambridge (1996)

Bruner, J.: Acts of meaning. Harvard University Press, Cambridge (1990)

Bruner, J.: Actual minds, possible worlds. Harvard University Press, Cambridge (1986)

Bruner, J.: Going beyond the information given. Norton, New York (1973)

Bruner, J.: Toward a theory of instruction. Harvard University Press, Cambridge (1966)

Campbell, D.T., Stanley, J.C.: Experimental and quasi-experimental designs for research. Rand McNally, Chicago (1963)

Chatterji, M.: Evidence on "What Works": An argument for extended-term mixed methods (ETMM) evaluation designs. Educational Researcher 34(5), 14–24 (2004)

Cobb, P., Confrey, J., diSessa, A., Lehrer, R., Schauble, L.: Design experiments in educational research. Educational Researcher 32(1), 9–13 (2003)

Connelly, F.M., Clandinin, D.J.: Narrative inquiry. In: Green, J.L., Camilli, G., Elmore, P.B. (eds.) Handbook of Complementary Methods in Education Research, pp. 477–487. Lawrence Erlbaum Associates, Mahwah (2006)

Cook, T.D., Means, B., Haertel, G.D., Michalchik, V.: The case for randomized experiments. In: Haertel, G.D., Means, B. (eds.) Evaluating Educational Technology: Effective Research Designs for Improving Learning, pp. 15–37. Teachers College Press, New York (2003)

Daly, J.A., Miller, M.D.: The empirical development of an instrument to measure writing apprehension. Research in Teaching of English 9, 242–249 (1975)

Frederiken, N.: Implication of cognitive theory for instruction in problem solving. Review of Educational Research 54(3), 363–407 (1984)

Geertz, C.: The interpretation of cultures. Basic Books, New York (1973)

Gibbons, A.(n.d.): Model-centered instruction and design layering (2011), `http://tip.psychology.org/gibbons.html` (accessed December 19, 2011)

Gibbons, A.S., Bunderson, C.V., Olsen, J.B., Robertson, J.: Work models: Still beyond instructional objectives. Machine-Mediated Learning 5(3&4), 221–236 (1995)

Goldman, R.: Video perspectivity meets wild and crazy teens: A design ethnography. Cambridge Journal of Education 34(2), 157–178 (2004)

Goldman-Segall, R.: Points of viewing children's thinking: A digital ethnographer's journey. Lawrence Erlbaum Associates, Mahwah (1998)

Lansiquot, R.D., Blake, R.A., Liou-Mark, J., Dreyfuss, A.E.: Interdisciplinary problem-solving to advance STEM success for all students. Peer Review 13(3), 19–22 (2011)

Lansiquot, R.D.: Making the virtual real: Using virtual learning communities for research in technical writing. In: Daniel, B.K. (ed.) Handbook of Research on Methods and Techniques for Studying Virtual Communities: Paradigms and Phenomena, pp. 224–232. Information Science Reference, New York (2010)

Lansiquot, R.D.: Advanced technical writing: Blending virtual communities (Special issue on blended learning). The Journal of the Research Center for Educational Technology 5(1), 57–63 (2009)

Lansiquot, R., Perez, M.: A student's guide to virtual worlds. Poster Session Presented in WCEMHT, Honolulu, HI (June 2009)

Lave, J., Wenger, E.: Situated learning: Legitimate peripheral participation. Cambridge University Press, Cambridge (1991)

McCandliss, B.D., Kalchman, M., Bryant, P.: Design experiments and laboratory approaches to learning: Step toward collaborative exchange. Educational Researcher 32(1), 14–16 (2003)

Padmanabhan, P.: Exploring human factors in virtual worlds. Technical Communication 55(3), 270–276 (2008)

Reed, W.M., Burton, J.K., Kelly, P.P.: The effects of writing ability and mode of discourse on cognitive capacity engagement. Research in Teaching of English 19(3), 283–297 (1985)

Schmid, R.: Real text in virtual worlds. Technical Communication 55(3), 277–284 (2008)

Spiro, R.J., Feltovich, P.J., Jacobson, M.J., Coulson, R.L.: Cognitive flexibility, constructivism, and hypertext: Random access instruction for advanced knowledge acquisition in ill-structured domains. Educational Technology 31(5), 24–33 (1991)

Spiro, R.J., Jehng, J.: Cognitive flexibility and hypertext: Theory and technology for the non-linear and multidimensional traversal of complex subject matter. In: Nix, D., Spiro, R. (eds.) Cognition, Education, and Multimedia. Erlbaum, Hillsdale (1990)

Suárez-Orozco, M.M., Qin-Hilliard, D. (eds.): Globalization: Culture and education in the new millennium. University of California Press, Berkeley (2004)

Tashakkori, A., Teddlie, C.: Mixed methodology: Combining qualitative and quantitative approaches: Applied Social Research Methods Series, vol. 46. SAGE Publications, Thousand Oaks (1998)

Vygotsky, L.S.: Mind in society: The development of higher psychological processes. Harvard University Press, Cambridge (1978/2006)

Weiss, C.H.: Evaluation, 2nd edn. Prentice Hall, Upper Saddle River (1998)

Yin, R.K.: Case study methods. In: Green, J.L., Camilli, G., Elmore, P.B. (eds.) Handbook of Complementary Methods in Education Research, pp. 111–122. Lawrence Erlbaum Associates, Mahwah (2006)

Abbreviations

ETMM	extended-term mixed methods
LSL	Linden Scripting Language
MCI	Model-Centered Instruction
MW	Mega watt
NEA	Nepal Electricity Authority
SL	Second Life
ZPD	Zone of proximal development

Chapter 12
A Smart Home Lab as a Pedagogical Tool

Antonio Sanchez and Lisa Burnell

Texas Christian University
Box 298850, Fort Worth, Texas, USA
{a.sanchez-aguilar,l.ball}@tcu.edu

Abstract. Active learning engages students in the development of knowledge and skills within a supportive environment. The domain of "Smart Homes" creates such a setting for this experiential learning. Smart Homes are intended to anticipate and meet inhabitant's needs as they unobtrusively adapt to changing preferences and goals. The vision is to create the optimal balance between current inflexible, limited systems and powerful, but complicated and unproven, pervasive systems. Finding this balance involves investigating approaches for decision-making, adaptation, knowledge representation, new user interfaces, and more. Within this rich and realistic domain, students are motivated to actively explore and learn. In this chapter, we describe our experiences with students over the past decade. We argue that one must move from virtual reality simulations to more physical environments, which we call real virtuality.

12.1 Introduction: An Intelligent Environment

12.1.1 A Smart Home Environment as a Learning Space

Learning is a combination of study, reflection, and experimentation. Following the dictum of constructivist educators, any process of knowing and learning begins with the individual. Offering individualized educational experiences that are both useful and appealing acts as a powerful motivator. This is the guiding principle for our use of a smart home environment to teach computer science concepts, especially adaptive learning systems. For the past eight years this environment has provided a unifying theme for student projects and faculty research. In this chapter, we present this multi-disciplinary, multi-institutional endeavor, accomplishments, and lessons learned.

Industry needs more computer scientists, a fact not helped by the rapid fall in enrollment. Though the trend is improving, it still is far below the peak in the early part of the century. There are many explanations for this discussed in the literature. In this chapter, the focus is on an active approach to computer science education intended to increase student interest and capability.

A. Peña-Ayala (Ed.): Intelligent and Adaptive ELS, SIST 17, pp. 293–314.
springerlink.com

Intelligent environments offer a meaningful learning venue in which to conduct experiments. Smart Homes, and related intelligent spaces, continue to entice us with their promise of anticipating and meeting our needs as they unobtrusively adapt to our changing preferences and goals (see Fig. 12.1). To date, the delivery of this promise has met with limited success in terms of functionality and consumer acceptance leaves room for opportunity. Many commercial and academic efforts are in progress to create true smart home systems, and, to a lesser extent, to understand what customers really need. Opportunity lies in the high-level reasoning necessary to exploit the next generation of smart home devices. Specifically, we seek to find the correct automation balance between today's comprehensible systems of limited flexibility and potentially powerful (but unproven) autonomous systems that are error-prone and complicated. Finding this balance involves investigating approaches for decision-making, adaptation, representing domain-specific knowledge, and new user interfaces.

Historically, homes are the places where people spend most of their private lives and it only makes sense to apply technological discoveries there. Electrical light and appliances were a prior generation's smart home (Bryson 2010). The birth of the X10 communication standard was first released in 1975. Today this standard and newer, improved protocols can transform a house into a Smart Home. A Smart Home can be considered the integration of various automated services to enhance the lives of its inhabitants. The integration is done mainly by software control of hardware available in the home using an internal communication system within the home. An important aspect using new technology is to develop and use new models and programming interfaces

Fig. 12.1 The Crescent Smart Home Lab at TCU

The approach we take in our smart home lab is eclectic in nature since it is mainly designed as a learning aid for our students. Moreover, this environment that entices them to get involved in AI and related research. Fig. 12.1 provides an image of our smart home lab, a 600 square foot one bedroom apartment equipped with appliances, plumbing, and home electronics, all of which can be accessed and control by software.

Students are given the opportunity to work individually and in teams on complex, open-ended problems, under the mentorship of the faculty. As reported earlier (Burnell et al. 2006), students can choose to work on smart home projects in their capstone design course. As a pedagogical tool this environment helps reach our educational goal: Engagement of undergraduate students in useful and state-of-the-art AI research. Computer Science and Computer Information Technologies students are most represented, with a few others from Electrical and Industrial Engineering. One goal of the lab is increased interdisciplinarity, especially recruiting students from Psychology, Health Care, and Engineering.

12.1.2 From Virtual Reality to Real Virtuality

Numerous virtual reality research show their results through computer simulated environments that present a quasi realistic situation. This emphasis has taken us away from real sensory/actuator data. Most current virtual reality environments are primarily visual experiences, presented through digital displays. The term Real Virtuality has been used to recognize the ever presence of virtuality in our daily life. This multi-sensory virtual environment implies full immersion (Castells 1996). In this chapter, we depart from a simulated world, asserting that real virtuality should address just the opposite: a path back to the physical world can comes from a real physical environment that virtually represents real world applications. Such is the case of the smart home lab, where the students experience the physical environment they are likely to find in a smart home. While a virtual home can simulate physical forces, it is not until students are confronted with a real door or a window that they get to measure the need to have a sensor and an actuator to apply a physical force. The idea was taken from (Negrete 2005). This is our approach to real virtuality. Operating in a simulated world is not the same as operating in reality (yet).

The creation of a physical lab experience maps to reality with high fidelity, providing access to real objects, and operating real actuators. In this physical environment students need to sense physical quantities such as movement, noise, temperature, fire, humidity, gas leaks, and the like and need to use actuators to operate appliances and devices such as fans, lamps, extinguishers, and cameras.

This is exactly the core of the chapter: understanding the implications of AI in daily life requires a rich learning environment that allows formation and testing of hypotheses.

Students must face implementation issues they may not otherwise (in a simulation or finely crafted homework assignment). While the lab is not a real home, it physically represents one, as opposed to virtually doing so (see Fig. 12.2). This approach does not undermine the benefit of using newer interface styles, like gesture-based mobile hardware controller. In fact the design and programming of intelligent displays and interfaces are very much part of students' work, as will be shown. Consider the wireless remote control that initially appeared for entertainment that can be used to control any appliance. The important paradigm shift is taken from delayed to instant gratification. Here lies the issue of design: to program a centralized, adaptive interface as a natural element in a smart home. We address this concept in some of the student projects.

The design and development of Smart Homes has attracted researchers of diverse disciplines from engineering to social sciences and architecture. A Smart Home is about creating innovative technology of value to inhabitants.This poses a challenge in interface design that promotes its use. It is here that the use of ambient displays and wireless communication controls needs to be addressed as part of this real virtuality. Through rich sensory physical experiences, the students gain a better understanding of the final products.

12.1.3 Overview of the Chapter

The next section discusses related work on Smart Homes as an example of intelligent environments and their use both in academia and in industry. The important niche of elder care is briefly addressed. Section three covers the various projects that have been developed by our students in the last years. In order to stimulate the application of ideas, some of the projects progress from conception to implementation.

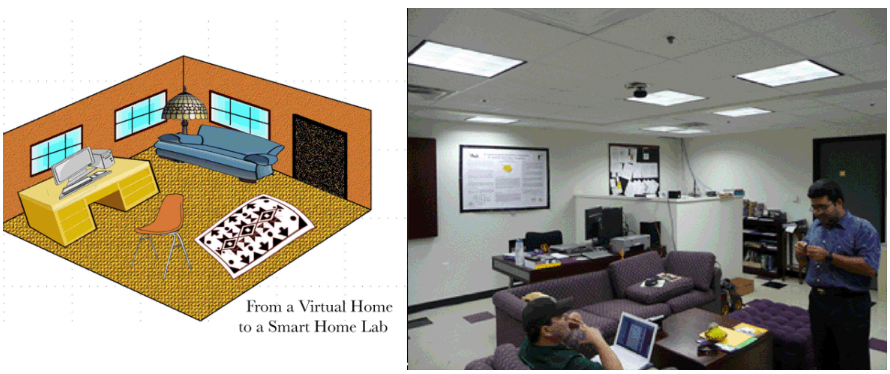


Fig. 12.2 From virtual reality to real virtuality

Because student teams are invited to propose and implement projects, diverse approaches have been employed to provide intelligence in a smart home. The lab offers an open space for students to test their ideas as long as they can come up with a reasonable design and implementation of a prototype that bears relevance to a smart home theme. Because some of these projects require hardware and other infrastructure purchases, we give some of our funding sources. Section four discusses the multi-university multidisciplinary international collaboration in the lab, with some guidelines for joint student research projects. Additionally, we explain how we have helped students in their individual research, in preparing it for presentation, and our take on helping undergraduates seek graduate studies in top universities. Prior to the final conclusions, we evaluate the pedagogical value of our activities.

12.2 Intelligent and Adaptive Educational-Learning Systems and Environments

12.2.1 Related Work

We present an alternative view as to what constitutes intelligent and adaptive learning, so here we mention standard approaches. Traditional computer aided instruction employ learner and topic models to provide automated teaching of relatively constrained skills. Much of this work has been greatly beneficial, whether helping a grammar student or a combat soldier. An Intelligent Learning System acts as a tutor for a topic area, e.g., Algebra. Sometimes referred to as CAI, or Computer Aided Instruction, these systems act as a teacher might, evaluating a student's performance and learning style, then providing customized exercises to target areas of weakness.

When teaching AI we have always sought a balanced mixture of theory and practice. Mastery of theoretical topics such as logics and adaptive learning are demonstrated, in part, through implementation (Luger and Stubblefield 2009, Heaton 2005). However students need to go beyond programming algorithms and confront real life needs, which are often too complex for them to build significant systems. Instead, we use our own experience to manually define tasks for students that we customize for their skill level. Part of this experience comes from new student teams working on the same projects as their predecessors. From this, case studies can be developed to create a lessons learned repository. This in turn informs us as to how topic and learner models may be developed so that some of our teaching may eventually be automated.

12.2.2 Smart Homes: Adaptive Educational-Learning Environment

The use of a smart home lab as an educational adaptive learning system also provides many options for research. We review here some of these research areas. First, consider basic home electrical wiring and the available communication protocols using such wiring. Although this chapter is not focused on electrical connections, student projects need a way to communicate with sensors and actuators. The use of internal wiring to automate a home has become popular for its ease of use and low cost. Table 12.1 (adapted from Briere and Hurley 2007), presents a list of commercially available protocols and products. One of the oldest and most developed is the X10 protocol released in 1975. Although it is slow and sometimes unreliable its wide spread deployment makes it a good the starting point. Using X10, a system can be installed to detect and trigger alarms and appliances. Students can get experience in modular architectural design by having to integrate newer protocols and products, e.g., Insteon, the Xbox Kinect, and IRobot, into the lab's existing smart home system.

The essential element in the Smart Home is the software to define the behavior of the various devices and not just the electrical wiring connections. Here is where reasoning and adaptive learning takes place and where we have focused most of our projects. Learning can take place based on the acquisition and application of knowledge about the smart home and its inhabitants. MAVHome and CASA (Cook et al. 2005, Cook et al. 2009) are representative of multi-disciplinary research projects dedicated to the creation of an intelligent home environment. In these systems, knowledge is acquired through multiple sensor data and questionnaires from selected inhabitants. Their approach deals with getting information on the activities of older adults during everyday activities. Using this information, inhabitants are unobtrusively monitored and rendered aid as needed. Sophisticated adaptive learning techniques provide a viable approach for the task.

Table 12.1 Commercial electrical wiring protocols

Protocol	Description	Reference
X10	Economical two way (sensor/actuator) control of up to 256 devices	www.x10.com
UPB	Universal Powerline Bus. Two way control up to 40 times faster than X10.	www.pulseworx.com
Insteon	Compatible with X10 faster also wireless accessible, faster and more reliable	www.smartlabs.com
ZigBee	Wireless two way control standard with early products being released	www.zigbeee.org
ZWave	An alternative to ZigBee, a fast wireless mesh where products are slowly being delivered	www.zen-sys.com
LonWorks	Industrial protocol beyond the needs of a home. Highly reliable	www.echelon.com

The most commonly reported case of smart home technology is in the area of elder care. Goals are to allow elders to remain in their homes, to reduce health care costs, and to provide caregiver services that may be unavailable. Elders report that they have a better quality of life within their own homes provided that they have some support. This presents a clear niche of opportunity. Aging is often accompanied with health problems that often lead to a decrease in quality of life. To live with autonomy in their homes, they need assistance with daily activities in different ways (Consolvo et al. 2004). They may need help to complete an activity, or to be warned when facing risks associated with performing an activity. As they suffer from natural cognitive impairment due to aging, they may forget events, situations or tasks. Recently, research on ontologies has been used to define scenarios within the context of a smart home (Rodriguez et al. 2008, Saldaña-Jimenez et al. 2009). The resulting ELDER ontology is being used in our projects.

Research on Smart Home technologies is a very active area of research, but again a distinction must be made between commercial products and research. Of the commercial projects, the AmigoProject (Janse 2008) is moving toward enhancing the quality of life based on common appliances. Conversely, some other systems can be little more than a proof of concept, leading to consumer confusion (see Table 12.2).

Because many homes have multiple inhabitants, there exists a need for social awareness in a smart home (Harper 2003). Thus, smart homes need a flexible design adaptive to different needs. Inhabitants may have different desires for thermostat settings. Some activities, such as laundry, are ones for which automation is not straightforward. Ownership of the various tasks is complex and not completely delegated. Some home activities are location independent, i.e., can be performed anywhere in the home. One might think that a WiFi remote control may solve this issue but sometimes the needed action goes beyond a simple process trigger. An interesting point is the distinction between a home chore (doing the laundry) and a desire to relax (hearing music). Most people are aware that free time is a scare resource when dealing with home chores and therefore the meaning of smartness lies in both the needs and the whims of the inhabitants. In a rather critical view of cyber technology, Agree argues that our daily activities go well beyond a data transaction and we have lost part of that reality (Agre 2002). For example to put it bluntly banking is more than going to an ATM, banking is a social activity and we must be aware of that. In the Smart Home context, we should consider this important conception.

Table 12.2 Smart Home classification

Level I : Homes which contain intelligent appliances
Level II: Homes which contain intelligent communicating appliances
Level III: Connected Homes
Level IV: Learning Homes
Level V: Attentive Homes

Ergonomic and cognitive issues for ambient displays benefit from architecture and design professionals. Beauty and elegance are as important in product design as is efficiency since people use a technology both for its looks and its usability. A good example of this is the success of the iPod/iPhone/iTouch/iPad line of products from Apple. This final point tell us that if products are aesthetically pleasing with reasonable utility then technology will keep changing our conception of a home.

This is the road to innovation. Based on level of intelligence, Smart Homes have been classified (Aldrich 2003), as shown in Table 12.2 Some claims can be exaggerated making the understanding of issues a bit difficult. Aldrich's classification validates our goal for focusing on the following EE and CS technologies: (1) Sensor Networks and Multimedia Computing, (2) Pervasive and Mobile Computing, (3) Robotics, (4) Middleware, (5) Ambient Display and Interface Design, (6) Bayesian Reasoning, Knowledge Representation, Hidden Markov Models, (7) Image and Speech Recognition, (8) Multi-agent Architectures and Systems, (9) Autonomous Computing, and (10) On demand or sense-and-respond computing.

Current intelligent appliances are at Level II, generally due to the lack of standardized communication among the various appliances within the home. We consider that automation in a smart environment should be viewed as a continuous cybernetic feedback loop with four main components as depicted in Fig. 12.3. The components that we expect in future smart home environments include those described in Table 12.3.

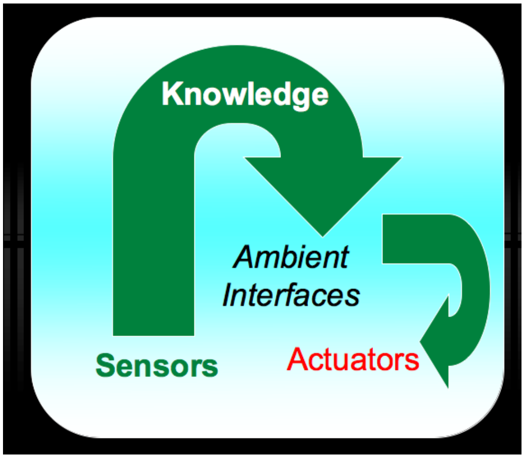


Fig. 12.3 Feedback loop in a Smart Home

Table 12.3 Intelligent environment functionality using CS/EE technologies

Sensors	Perceiving the state of the environment by means of multiple sensors
Ontologies	Reasoning about the state together with task goals and outcomes of possible actions using a defined ontology
Actuators	Acting upon the environment to change the state by means of a set of simple but precise and accurate actuators
Ambient Interfaces	Taking a broader view of a GUI by making use of the entire physical environment as an interface to digital information.

12.3 Research and Faculty Mentoring

12.3.1 Research Opportunities

In this section, we discuss projects developed by our students in the smart home. These are of an eclectic nature due to the fact that it is mainly designed as a student learning aid. Moreover, it is precisely this type of environment that entices students to get involved in AI research. We are committed to the development of a learning space for students; a space that allows them to be creative problem solvers who communicate and work with others. In this context students are given the opportunity to work individually and in teams on complex, open-ended problems, under the mentorship of the faculty. It is important to keep students engaged and this requires that they feel appropriately challenged by interesting activities. To fulfill this double mission we invite graduate students to be more purposive (i.e. goal definers), while the undergraduates are guided to be more purposeful (i.e. goal achievers). The research opportunities available in our smart home lab as we present to our students are summarized in Fig. 12.4.

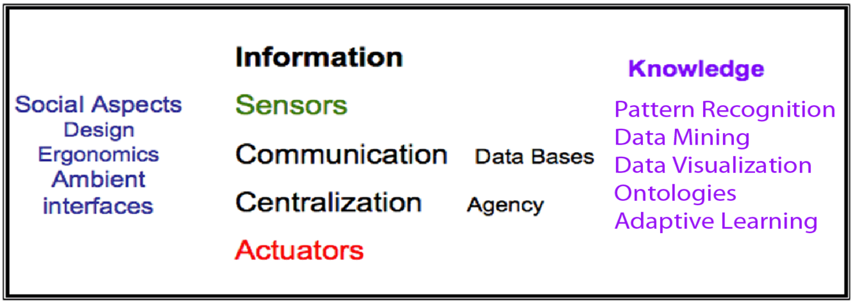


Fig. 12.4 Smart Home research opportunities

12.3.2 SDBI Architecture

We begin our discussion with a project that combines three of the four elements in the Smart Home feedback loop discussed earlier. The project was named SDBI (Selecting Digitally Binary Inputs) and was developed by two graduate students during a summer internship at our lab. The general framework of the SDBI architecture is designed to operate within a home environment, sensing the environment and inhabitants, reasoning about the appropriate actions, and responding accordingly as depicted in Fig. 12.5. Using an elder care application scenario, a smart home should respond to the needs and disabilities of elderly inhabitants to prevent injury or death. Additionally, the home should provide commonly available capabilities to aid the inhabitants and their caregivers, e.g., health care appointment reminders, medication monitoring, inventory management of basic necessities like food, and remote monitoring. We will focus on the safety monitoring components in this chapter, for which the system needs to sense the environment, determine risk, take some immediate actions, and notify caregivers. The SDBI software architecture is based on the following five components: (1) Sensor Information Acquisition, (2) Sensor Communication, (3) Sensor Information Fusion, (4) Decision Model and Ontology, and (5) Actuator Controllers.

Elder care requires sophisticated sensor processing and while others have made significant contributions in this area, our interests center on knowledge representation and reasoning. Therefore, we have restricted the lab environment sensors to detect the following: (1) relative humidity, (2) ambient temperature, (3) movement, (4) natural gas, and (5) air quality control such as smoke, CO_2, CO, and other pernicious gases.

SPOTs (Small Programmable Object Technology) establish bidirectional communication with the server and unidirectional with the required services (Fig. 6). If the Base-Station finds at least one of the required services, it sends a Services Request to the Free-Range SPOT. This student-friendly hardware with software tools are open-source and familiar to the student researchers. SPOTs allowed the students, already skilled in Java programming, to experiment with communication protocols. All sensor data are transmitted to the decision model before any actuation occurs.

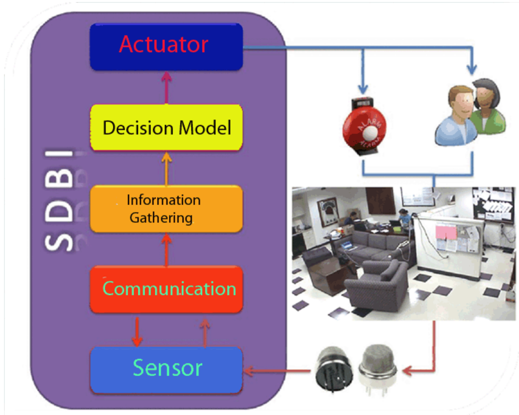


Fig. 12.5 The SDBI system architecture

Having the faculty choose system development tools and platforms reduces the risk of overwhelming the undergraduate students. This is perhaps one our biggest lesson learned, and as most are, a lesson that seems obvious now. It is easy to get allured by the promise of the "next big thing".

Protégé, an OWL compliant language (Herman 2007), is used to define an ontology that relates sensor features and actions (see Table 12.4). The ontology and rules define concept descriptions and their relationships, possible actions, and environmental parameters (sensor inputs). A simple scenario is one in which the inhabitant gets distracted while cooking something on the gas stove. A fire starts. Detecting a rapid change in temperature and smoke in the kitchen, the system decides there is a high risk of a fire, so it sends a notification to the fire department and the caregivers, and activates exhaust fans and a fire-suppression system.

The parameters are evaluated as the rules are parsed by the system. By defining basic Jess files such as: "burning.clp", "ignition.clp", "intoxication.clp" and "cigar.clp" different parameters are processed. Using the actual values the decision module sends to the Actuator Controller a set of commands as described in a table of risks. New rules can be incorporated manually or via learning patterns. The action module controls the different actuators (e.g. alarm, sprinkler, lamp, cameras). In order to use commonly available infrastructures like electrical wiring, a "Fire-Cracker" generates signals received by X10-controlled electrical appliances. By programming the various ports and devices a simple actuator interface can be readily implemented (see Fig. 12.6). The system is fully functional and various scenarios have been tested. The current GUI of SDBI is given in Fig 12.7.

Table 12.4 Some of the common household risks

Rule	Risk	Notification
IF natural gas and inhabitant motion is detected	CO poisoning, fire, explosion	Turn on exhaust fans, extinguish open flames, and notify inhabitants and caregivers.
IF smoke and a rapid change in ambient temperature that exceeds 100° F is detected	Fire	Turn on lights and exhaust fans, notify inhabitants and caregivers.
IF cigar smoke is detected	Unsafe inhabitant behavior	Warn: "Do not smoke inside the house"; notify caregiver.

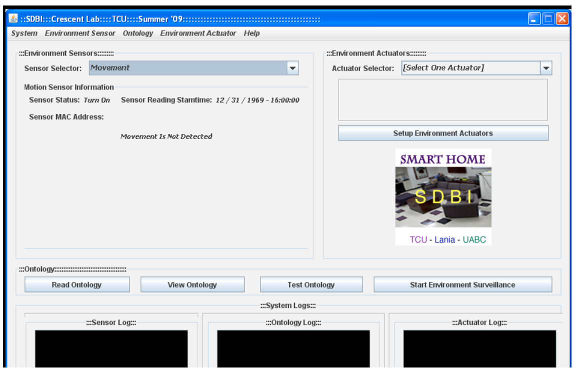


Fig. 12.6 SDBI general GUI showing sensors and actions

Fig. 12.7 The SDBI graphic user interface

The SDBI follows a Model-View-Controller (MVC) architectural design pattern to integrate diverse hardware and software components and to allow for iterative, incremental development (see Fig. 12.8). Using Java as the primary language for all lab projects eases system integration tasks and teaches software engineering principles.

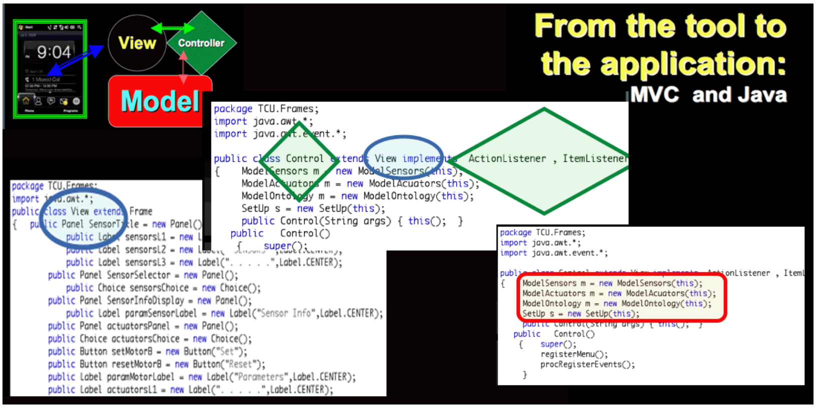


Fig. 12.8 The MVC architecture and corresponding Java code

12.3.3 Smart Entertainment Controller

The Smart Entertainment Controller (SEC) controls audio/visual equipment, temperature control, and lighting. Inhabitants can record and save favorite device settings. A modest learning capability based on TV viewing behavior can suggest programs that might be of interest. The client-server architecture communicates via SOAP messages. Important learning objectives are machine learning, distributed computing, middleware, and multi-modal interface design.

Clients provide the interfaces to the server in order to perform tasks. Currently implemented clients have speech and graphical interfaces. Some of the named pattern scenarios are shown in Table 12.5. The inhabitant has the ability to modify the scenarios and to define new ones. A simulator was also created to allow students to test SEC software.

Table 12.5 Some pattern scenarios for the Smart Environment Controller

Pattern	Trigger	Tasks
Work Prep	- User command - Alarm goes off - Sensor (wakeup)	Adjust thermostat Coffee maker on Adjust shower water setting
Leave Home	User command, sensor or prediction	Adjust thermostat
Bed time	User command	Turn off lights Turn on music in bedroom
Watch movie	User command	TV and DVD on Movie lists on TV screen

Communicating technical topics is another learning experience, as is evidenced by one Senior Design team describing the project in a Student Research Symposium:

"The SEC simulator is a software environment designed to simulate the management of media and environmental devices in the room as well as simulate inhabitants. It gives current status of devices that have been installed. The Simulator functions by querying the device software running on the system. It is not required for the SEC to function properly; it merely shows the user the result of functions carried out by the Controller. The simulator's primary use is in development and testing."

The SEC simulator requires a DeviceID_Simulator file for each device installed on the system. This is the portion of the device code that will indirectly receive messages from DeviceID_System to keep the simulator image and device status data up to date. Simulator functionality is provided for development purposes and can easily be removed. Currently, it allows the device files to be written and tested without building actual hardware.

Important pedagogical topics include user interface design, simulation, hardware/software integration, multi-process communication, and like the others, software engineering principles.

Students must provide for the definition of constraints, preferences, and priorities, some of which are based on inhabitant behavior prediction. For example, water usage priorities may be defined as {shower, dishwasher, washing machine, and sprinkler system}. Exceptions may include not running the sprinkler system if the *expected* shower time is within the next 30 minutes. Accounting for multi-inhabitant household presents constraint processing challenges. For example, if the wife prefers warmer temperatures in the living room, and the husband, who prefers lower temperatures, enters the room.

12.3.4 Entry Recognition and Audio Interface Using Agent Communication

Movement recognition is an easy task to perform using a simple sensor; however, for the next project we wanted to have face recognition to determine whether a human had entered the room and to eventually determine which person had entered. For this case using a Java version of the ALISA (Bock 1998) engine for image recognition and segmentation and SPHINX (Lamere 2004) for speech recognition our next group of students were able to design an entry recognition system that could detect if a person entered the home and then speak to the person to establish an initial communication. An agent system connects the image software with the speech recognizer and synthesizer (see Fig. 12.9). The user can have a simple conversation with the system that when in trouble searches *Wiktionary* for words or names not found in its basic database.

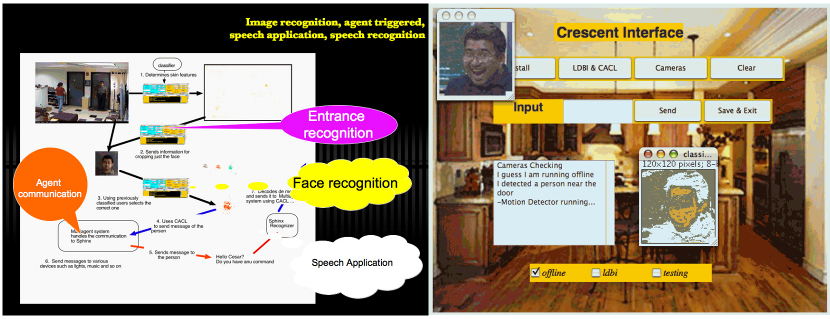


Fig. 12.9 Image entry recognition and final result

12.3.5 Speech Recognition and Virtual Human Interfaces

Another project does speech recognition with a modified version of SPHINX. An undergraduate student used a context based decoding algorithm designed to iteratively prune the speech recognizer's vocabulary based on the initial hypotheses of previous passes.

The inclusion of this enhancement in the smart home provides another example of adaptive learning using HMM algorithms implemented in SPHINX. Adding this module provides another level of sophistication to a smart home since an overall system could include it with the rule base system and ontology as discussed elsewhere.

A virtual human interface has been progressively developed by Senior Design project teams. In the earliest version low resolution images were used to test the interaction with a rule-based pattern-matching response system, developed using the Virtual Human software (Plantec 2004). A newer version was later created that has a much higher quality image and background, as well as full integration with the Smart Kitchen. The Senior Design projects have demonstrated successful examples outside the smart home environment using the same hardware and software technologies described herein. For example, students were able to combine the use of SPOTs with HMMs algorithms to train a system to recognize gestures (Stromberg et al. 2010). In addition, students developed a practical application to assist fitness trainers assess and track clients.

12.3.6 Enhanced Simulation Environment for Testing and Teaching Advanced Concepts

The SimHouse software was created for use in the Smart Home technology course and for use in testing smart home components. It was also used in the networking and database lab exercises in the course. SimHouse is a sensor/actuator simulator that allows one to simulate input and verify output of higher-level smart environment components like active databases and reasoners. Unlike the SEC simulator described earlier, SimHouse provides simulation capabilities for more than testing entertainment devices and lighting. It is built on a client–server model where the simulation of sensor banks and actuator systems act as services to the client components under test using the SOAP protocol. SimHouse is a session-based simulator that supports multiple users, simultaneous simulations of different environments and environment states, and event logging. The primary driver of a SimHouse simulation is a predefined scenario which sets the initial state of the sensors and actuators, and then allows these states to change either by direct scenario input or as a result of temporal conditions (e.g. start a fire in room A at time T) stored in the scenario file. Sensors and actuators include HVAC control, lighting, kitchen appliances, entertainment devices (TV, DVD, etc.), and sensors for temperature, motion, and light. In addition to accepting changes based on the scenario file input, SimHouse also allows session data to change based on the connected client's modification of the environment through actuators and sensor retargeting.

12.3.7 Ambient Displays and Interfaces

The final student project we present is a multi agent system that integrates speech recognition, a causal model, and the construction of an ambient display of light and a sound sensor circuit connected through a parallel port. The prototype was developed again in Java by team of international graduate students. Again the developed HCI application was designed to address the needs of the elder population. The Ambient Display has a bidirectional communication to send and receive information from the agent at the server end. With the use of a simple thermistor the circuit determines the water temperature and converts it into digital signal. Fig 12.10 depicts the conception of the sensor application; the illustration clearly shows the purpose of its design. The ambient display system uses a red or a blue light to characterize the temperature of the water and a voice to alert its users. It must be stated that while this type of products may already exist in the market they are closed circuits and are not open to be connected to a centralized computer system, thus making it difficult to control them from a far.

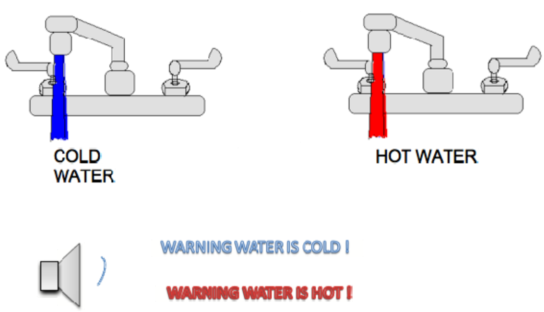


Fig. 12.10 Ambient display conception

The system uses the ELDeR ontology previously described (Saldana-Jimenez 2009) and its design to determine the risk associated with the daily activities at the home using a Bayesian causal model. Furthermore, the model is used to learn the user temperature preferences and alert accurately.

Fig. 12.11 shows the water display deployment implemented by our able students. The top control is used to obtain the information surrounding the faucet and communicate with the centralized server. The lower control converts the digital information to tangible information for the user, in this case it may be used by the user be it an elder person or a nurse. The Ambient Display has a bidirectional communication to send and receive information from the agent at the server end. With the use of a thermistor the circuit determines the water temperature and converts it into digital signal, this is achieved by a reduction in the thermistor resistance in the presence of hot water and increases it with cold water, which sets the transistor in the points of cut or saturation; thus generating a signal with maximum and minimum voltage that can be interpreted as a digital zero or one.

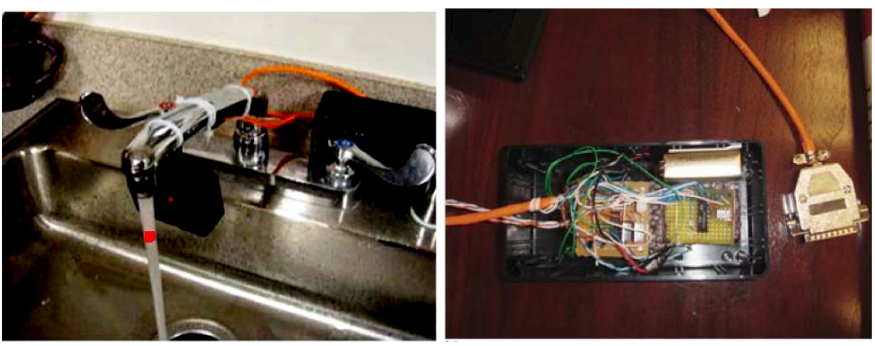


Fig. 12.11 Working prototype for the ambient display

12.4 Intelligent and Adaptive Education Learning Experiences

12.4.1 Relationship to Recommended Curriculums

The projects that have been developed correspond to a number of knowledge areas suggested in the body of knowledge required in high quality computer science programs, and specifically dealing with artificial intelligence while Smart Homes are considered to be in the domain of pervasive or ubiquitous intelligent environments. The ACM has stressed the point "increasingly computing students gain employment in application areas. To be effective and to be able to play a leadership role this often entails gaining some sophisticated domain knowledge" (ACM 2008).

As the need to update curriculums accelerates, our lab can be used as a testing ground for new topics and tools. New tools, like programming languages, can be introduced and evaluated. As an example, a 2nd or 3rd year student can be asked to port an existing program from one language to another. The initial language will be one in which the student is comfortable, like Java, and the target language will be new to the student, e.g., Python. The student not only has learned a new language, but also has been offered a window into the potential learning challenges. Students may face when the new language may incorporated into a course. While a course in Programming Language Concepts (PLC) exposes students to multiple programming paradigms, it does not generally do so using existing, relatively complex programs. PLC students commonly develop small programs in the target language. The former approach offers the ability to better assess implementation issues in prospective curriculum changes. Changing the first year programming language in a curriculum is difficult – training faculty, choosing multiple new textbooks, developing new lab exercises – so it is quite valuable to have actual experience with one's own student population. Finally, the lab can offer students learning experiences in non-core topics, e.g., Image Processing, that the department cannot offer as an elective because of faculty constraints or small student population.

12.4.2 International Multidisciplinary Collaboration

Students are actively engaged in dissemination of their research. Every spring students are invited to present a poster in the Student Research Symposium (SRS). All students and faculty from the various departments of the College of Science and Engineering attend the event.

Computer Science students have presented their work at symposiums and conferences including the SRS, ACM's SIGCSE Conference, the National Conference on Undergraduate Education, and the North Texas Area Student Conference.

Summer internships for international graduate students were made available through research grants. This affords the opportunity for faculty and undergraduate student collaboration, something especially important in undergraduate only departments. We have maintained active research discussions with institutions in Mexico, such as Laboratorio Nacional de Informática Avanzada (LANIA), Instituto Tecnológico de Monterrey (ITESM), Universidad de Guadalajara and Universidad Autonoma de Baja California. Perhaps the best way to describe how successful this graduate internship has been is to mention that as these graduate students enter into the lab we simply ask them: Given the facilities you see here, can you think about a project that includes the development of a prototype? After spending a day in the lab, they have been able to conceive an initial idea and then they dedicate their summer to develop it. Our mentoring process consists of meeting with them at least weekly and providing any assistance they might need. We feel that the payoff of this international collaboration among faculty and students of international research centers has been the opening of new ways to achieve our academic goals.

12.4.3 Student Preparation for Graduate Studies

Since its conception an important part of our smart home lab project was to promote students to publish or collaborate in joint research publication, we believe that we have been successful at this as the reference show. All of our students have been exposed to the task of writing technical reports and in some cases they have been the first authors in some of the publications.

We have been able to guide our undergraduate students to pursue graduate school; in part this has been achieved by the collaborative environment of faculty, invited graduate and undergraduate students. Working with students from other institutions has turned out to be a good experience, our students learned to make new acquaintances and collaborate and have been able to turn in an integrated prototype application. We have learned that the continuous collaboration between the groups forged a close relation between them that will help the students beyond the scope of the summer project. The TCU undergraduate students have found the lab to be an alluring project and collaborated well with the team.

12.4.4 Measuring Pedagogical Value

Since the lab began in 2002, there have been approximately 25 student, student-faculty, and faculty publications on Crescent Lab research activities. Examples include (Burnell et al. 2006, Denkowski et al. 2007, Sanchez et al. 2009, Wong and Burnell 2004).

Approximately 25% of the Computer Science and CIT students have worked directly as research assistants, and an additional 20% were involved in Senior Design and Independent Study courses. There have been 8 visiting international graduate students, whose work in the lab has been a part of their PhD or Master's theses at their home institutions. A few research students have come from Electrical Engineering, with immediate plans for more of their students and some faculty.

Publication rates and graduate school attendance for lab research assistants are significantly higher than the general CS student populations at out University. Fig. 12 shows a graduate school attendance comparison. Students having worked in the lab attend graduate school more often than the general department student population. Moreover, within the lab, a higher percentage of students go on to pursue technical Ph.D. or Master's degrees. Moreover, Crescent Lab students publish three times as many articles than other students in the department.

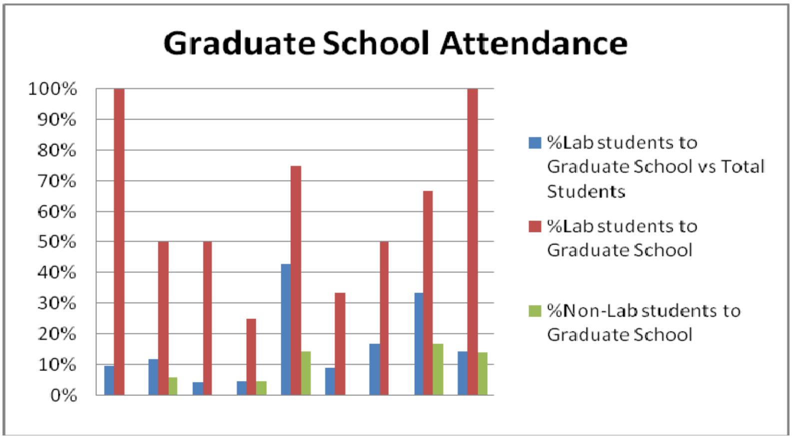


Fig. 12.12 Graduate school comparison

12.5 Conclusions

As educational institutions, universities have always played an important role in both education and its assessment. Yet these two activities have been so tightly coupled that they may appear difficult to separate. Indeed, it seems that we are both judges and parts of the process. This comes from a long history of European guild schools where universities originated. The master was the one to decide when the apprentice was ready for the world. However, today and outside of the academic world, when our students look for a job, they need to be able to self-assess their work. In the case of our lab, the concern of how learning took place to back seat; the main concern became performance while developing the projects.

Among the important lessons we have learned is that in any collaborative endeavor, especially one that includes expectations and pace of work of young minds can lead to miscommunication, failed deadlines, and rework. And, so research tasks suitable for undergraduate students must be clearly defined in advance. Undergraduate research assistants require considerable guidance, even among the very best students, meaning that faculty should view mentoring as a challenging teaching task. The results of our efforts have paid handsomely by providing our students with better preparation for graduate studies or industry jobs, as well as direct interaction with faculty who expect them to do more than assist the graduate students. For the case of graduate students this is not the case and our summer research internship allows them to define their own goals, while they develop a prototype that is later used by the undergrads. It is important to keep students engaged and this requires that they feel appropriately challenged by interesting activities. A benefit of this increased international research collaboration among faculty and students opens new ways to achieve our academic goals, specifically in Computer Science.

Believers in the commitment to undergraduate research, active learning, and individualized mentoring, we have found that the smart home research lab has been a rewarding experience for faculty and students. Many students that have worked in our lab go directly into recognized graduate schools (e.g., University of Chicago, CMU, and University of Colorado) while others find that their experiences are directly transferrable to the work environment. Our industry advisory board commonly bemoans the lack of "soft skills" (communication, managing stress under deadlines, and working in a team with diverse backgrounds, personalities, and experiences) that they see in many new graduates. By giving our undergraduate students an immersive experience, performing "real" research tasks alongside faculty and visiting graduate students, they graduate with many of the skills they need to take their place as professionals. Since we have presented an alternative view as to what constitutes intelligent and adaptive learning, let us here briefly mention some standard approaches; traditional computer aided instruction employ learner and topic models to provide automated teaching of relatively constrained skills.

Much of this work has been greatly beneficial, whether helping a grammar student or a combat soldier. An Intelligent Learning System acts as a tutor for a topic area, e.g. Algebra. Sometimes referred to as CAI, or Computer Aided Instruction, these systems act as a teacher might, evaluating a student's performance and learning style, then providing customized exercises to target areas of weakness. To conclude the chapter let us restate that although our approach is different from other educational tools that use various schemes of adaptation in order to provide novel pedagogical tools. We want to re-emphasis that the skills we are trying to teach are currently too complex to build significant automated systems, this is why we have created the smart home lab; the results reported here show that this could be a significant trend for computer science education.

Acknowledgments. This work was sponsored in part by grants from TCU and selected Mexican universities. We wish to acknowledge our TCU and international students, in particular Jorge Acosta, Hector Ceballos, Omar Cetina, Avinash Chandra, Michael Denkowski, Jonathan Clark, Cesar Garcia, Juan Pablo Garcia, Ahn Pham, Sneha Popley, Humberto Quijano, Diana Saldaña, Ricardo Tercero, Dimitar Zlakov, Edward Tran, Edwin Wong, and Vince Guerin.

References

ACM and IEEE-CS. Computer Science Curriculum 2008 (2008), `http://www.acm.org//education/curricula/ComputerScience2008.pdf` (accessed November 5, 2011)

Agre, P. E.: Cyberspace as American Culture. Science as Culture 11(2), 171–189 (2002)

Aldrich, F.K.: Smart homes: Past present and future. In: Harper, R. (ed.) Inside the Smart Home, pp. 17–40. Springer, London (2003)

Bock, P.: ALISA: Adaptive Learning Image and Signal Analysis. In: Proceedings of the SPIE Applied Imagery Pattern Recognition Conference, Washington, DC (1998)

Briere, D., Hurley, P.: Smart Homes for Dummies. Wiley, New York (2007)

Bryson, B.: At Home. Doubleday, NY (2010)

Burnell, L., Priest, J., Durrett, J.: Teaching Distributed Collaborative Software Development. IEEE Software Special Issue on Software Engineering Education, 86–93 (September /October 2002)

Burnell, L., Sanchez, A., Priest, J., Hannon, C.: The Crescent Lab: A smart home lab for students. In: Proceedings of ENC 2006, pp. 55–61. IEEE Computer Society, Los Alamitos (2006)

Castells, M.: Rise of the Network Society. Blackwell, Oxford (1996)

Comenio, J.A: Orbis Sensualium Pictus. Borbildung & Benahmung, Nuremburg Germany (1658); Available trilingual edition by Mexico: MA Porrua (1993)

Consolvo, S.P., Roessler, P., Shelton, B.E., LaMarca, A., Schilit, A., Bly, S.: Technology for Care Networks of Elders. IEEE Pervasive Computing Magazine: Successful Aging 3(2), 22–29 (2004)

Consolvo, S., Roessler, P., Shelton, B.E.: The CareNet Display: Lessons Learned from an In Home Evaluation of an Ambient Display. In: Davies, N., Mynatt, E.D., Siio, I. (eds.) UbiComp 2004. LNCS, vol. 3205, pp. 1–17. Springer, Heidelberg (2004)

Cook, D., Schmitter-Edgecombe, M., Crandall, A., Sanders, C., Thomas, B.: Collecting and disseminating smart home sensor data in the CASAS project. In: Proceedings of the CHI Workshop on Developing Shared Home Behavior Datasets to Advance HCI and Ubiquitous Computing Research (2009)

Cook, D.J., Das, S.K.: Smart Environments. Wiley-Interscience, New Jersey (2005)

Denkowski, M., Hannon, C., Sanchez, A.: Spoken Commands in a Smart Home: An Iterative Approach to the Sphinx Algorithm. In: Gelbukh, A., Kuri Morales, Á.F. (eds.) MICAI 2007. LNCS (LNAI), vol. 4827, pp. 1025–1034. Springer, Heidelberg (2007)

Flanagan, B.: Smart Home. Workman Publishing, NY (2008)

Goodwin, S.: Smart Home Automation with Linux. Springer-Apress, New York (2010)

Heaton, J.: Introduction to Neural Networks with Java. Heaton Research, St Louis (2005)

Herman, I.: W3C. Web Ontology Language (2007), http://www.w3.org/2004/OWL/ (accessed November 5, 2011)

Janse, M.D.: Amigo –Ambient Intelligence for the Networked Home Environment (2008), http://www.sigma-orionis.com/nesummit.eu/Presentations/Day2/SessionsB/B4/Amigo.pdf (accessed November 5, 2011)

Lamere, P., Kwok, P., Walker, W., Gouvea, E., Singh, R., Raj, B., Wolf, P.: Sphinx-4: A Flexible Open Source Framework for Speech Recognition. Sun Microsystems, Report Number: TR-2004-139 (2004)

Luger, G., Stubblefield, W.: AI Algorithms, Data Structures, and Idioms in Prolog, Lisp and Java. Addison Wesley, Reading (2009)

Meyer, G.: Smart Home. O'Reilly, Hacks Sebastopol (2005)

Negrete-Martínez, J.: Three Steps to Robo Sapiens. In: Ślęzak, D., Yao, J., Peters, J.F., Ziarko, W.P., Hu, X., et al. (eds.) RSFDGrC 2005, Part II. LNCS (LNAI), vol. 3642, pp. 727–733. Springer, Heidelberg (2005)

Northrup, T., Faulkner, E.: Home Hacking Projects for Geeks. O'Reilly, Sebastopol (2005)

Plantec, P.: Virtual Humans. Amacom, New York (2004)

Rodriguez, M., Curlango, C., Garcia, J.P.: An Agent-based component for identifying elders. In: Proceedings of the 3rd Symposium of Ubiquitous Computing and Ambient Intelligence, University of Salamanca, Spain (2008)

Saldaña-Jimenez, D., Rodríguez, M.D., García-Vázquez, J.-P., Espinoza, A.-N.: ELDeR: An Ontology for Enabling Living inDependently of Risks. In: Meersman, R., Herrero, P., Dillon, T. (eds.) OTM 2009 Workshops. LNCS, vol. 5872, pp. 622–627. Springer, Heidelberg (2009)

Saldaña-Jiménez, D., et al.: A Context-Aware Component for Identifying Risks Associated to Elder's Activities of Daily Living. In: Proceedings of IEEE PERVASENSE 2009, London, UK (2009)

Sanchez, A., Diaz, R., Bock, P.: A CLS Hierarchy for the Classification of Images. In: Gelbukh, A., de Albornoz, Á., Terashima-Marín, H. (eds.) MICAI 2005. LNCS (LNAI), vol. 3789, pp. 369–378. Springer, Heidelberg (2005)

Sanchez, A., Hannon, C., Garcia, J.P., Garcia, C., Ceballos, H., Cetina, O.: Service Robots on a Smart Home Lab. In: Proceedings WSB-7th MICAI, pp. 3–13 (2008)

Sanchez, A., Tercero, R., Saldaña, D., Burnell-Ball, L.: SDBI: An Ontology Based Smart Home Lab Environment. In: Proceedings WILE-8th MICAI, pp. 217–226 (2009)

Strebe, M.: INSTEON: Smart Homes for Everyone. IUniverse, NY (2009)

Stromberg, P., et al.: FROG. TCU SRS, FortWorth, TX (2010)

Wisneski, C., et al.: Ambient Displays: Turning Architectural Space into an Interface between People and Digital Information. In: Proceedings of the 1st IWCo Build (1998)

Wong, E., Burnell, L., Hannon, C.: An Active Architecture for Managing Events in Pervasive Computing Environments. In: Proceedings FLAIRS 2004 (2004)

Abbreviations

GUI	Graphic User Interface
HCI	Human Computer Interface
HMM	Hidden Markov Model
MVC	Model View Control Design
OWL	Ontology Web Language
SEC	Smart Entretainment Contoller
SDBI	Selecting Digitally Binary Inputs
SOAP	Simple Object Access Protocol
SPOT	Small Programmable Object Technology

Chapter 13
Supporting Hybrid Courses with Closed-Loop Adaptive Training Technology

James E. McCarthy[1], John L. Wayne[1], and Brian J. Deters[2]

[1] Sonalysts, Inc.
215 Parkway North, Waterford, CT, USA
mccarthy@sonalysts.com, waynej@sonalysts.com
[2] Center for Surface Combat Systems
5395 First Street, Dahlgren, VA, USA
brian.deters@navy.mil

Abstract. At the beginning of this millennium, the U.S. Navy was faced with a high-stakes course, aging equipment, and an academic drop rate that was becoming unaffordable. To combat these problems, they turned to advanced training technology that provided students with tailored instruction and an increased volume of supervised practice opportunities. In this chapter, we discuss this technology, how it was incorporated into the course, and the effect that it, along with other instructional interventions had on performance.

13.1 Introduction

With waves rocking the ship, the U.S. Navy air controller stationed in the ship's nerve center stares intently at her screen. She must help a helicopter participating in a search and rescue mission safely find its way back to the ship before it runs out of fuel or visibility decreases to the point that landing becomes dangerous. To make matters worse, the helicopter must stay clear of other aircraft participating in the rescue mission and avoid straying into the national airspace of a hostile country. As the clock clicks down and the pressure mounts, the air controller gratefully recalls the world-class training she had received a few months earlier.

Onboard U.S. Navy ships, Antisubmarine Warfare (ASW)/Antisurface Warfare (ASUW) Tactical Air Controllers (ASTACs) take over the fixed and rotary-wing aircraft that support Fleet operations against hostile submarines and surface ships.

The ASTACs must be qualified to provide not only mission planning, mission coordination, and tactical control of helicopters but also advisory control, positive control, and safety of flight for helicopters and fixed-wing ASW/ASUW aircraft.

A. Peña-Ayala (Ed.): Intelligent and Adaptive ELS, SIST 17, pp. 315–337.
springerlink.com

In this chapter, we examine the challenges faced by the U.S. Navy in training prospective ASTACs and how advanced training technology, including closed-loop adaptive training, was used to meet this challenge.

13.2 Problem Statement

Prospective ASTACs undergo a 10-week course of instruction (COI) at U.S. Navy schools in San Diego, CA or Norfolk, VA. Traditionally, the COI had three major components. Instructors provided fundamental knowledge and skill training during traditional stand-up lectures. Students then practiced the skills in a one-on-one simulation setting in which the student assumed the role of the ASTAC and the instructor assumed the role(s) of the aircraft flight crew, the shipboard ASTAC supervisor, and (as necessary) the submarine or surface ship being investigated. Hardware and personnel limitations often dictated that a given student would be afforded only limited practice opportunities. Periodically, written and practical examinations (using the same one-on-one simulation system) were administered.

As a capstone event, students were given an opportunity to control live aircraft. Some recognized deficiencies were associated with this legacy approach. First and foremost was the high academic drop rate associated with this course. ASTACs hold life-and-death responsibilities; as such, the standard for their performance is quite high. Unfortunately, with so few opportunities for practice, a high percentage of the students failed to maintain the expected level of performance and were dropped from the course. The drop rate for fiscal years 2000-2005 is summarized in Fig. 13.1. This rate includes all sources of attrition (*i.e.*, academic drops, medical issues, recalls to duty, *etc.*). Over this period, the drop rate for the ASTAC course of instruction averaged approximately 28%.

This high drop rate was compounded by a throughput limitation associated with the legacy training approach. For the ASTAC COI, throughput was severely limited by an antiquated training device. The training device, 20F18, was used to support practice within the course. During a practice session, a student would be paired with an instructor in a one-on-one arrangement. The student would sit at one console and act as an ASTAC. The instructor would sit at the instructor station and control the behavior of platforms in the scenario, role-play the ASTAC's shipboard supervisor and the aircrew on the controlled aircraft, assess student performance, and perform coaching. While some instructors and students worked together, the remaining students waited. This created a bottleneck in the curriculum and severely limited the number of student practice opportunities. In fact, it was not uncommon for students to only have one or two practice opportunities for each new skill.

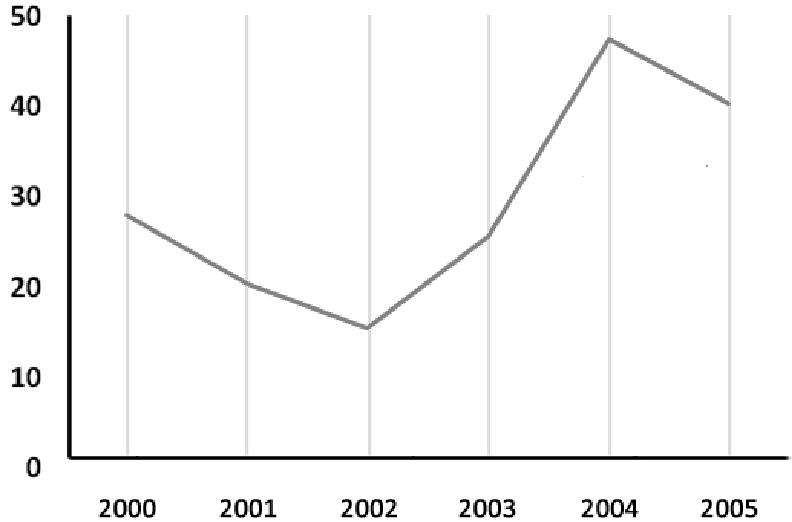


Fig. 13.1 ASTAC drop rate percentage by fiscal year (2000-2005)

Moreover, as the equipment aged, more time and money was devoted to maintaining it. In FY03, eight hours a day and $16,000 a month per site was devoted to maintenance of Device 20F18. In the best case, Device 20F18 could accommodate four students (and four instructors at any given time). However, one or more stations were typically down for maintenance, further curtailing throughput.

In the last half of the decade, the U.S. Navy made several improvements to the ASTAC COI:

- Replaced device 20F18 with a personal computer (PC)-based system,
- Employed a closed-loop adaptive training system to deliver ASTAC instruction and provide additional practice opportunities,
- Increased enforcement of pre-requisite completion,
- Established a one-week pre-screening/prep course,
- Implemented more rigorous academic review standards.

In this chapter, we focus primarily on the first two of these solutions – the use of advanced training technology to support instruction, practice, and assessment.

13.3 Solution Description

A multidisplinary team led by the three authors undertook a project to introduce advanced training technology into the ASTAC COI. The project had several goals. First, the team wanted to replace the costly legacy training device with a more cost-effective PC-based training system. Second, the team wanted to increase throughput by increasng the amount of time that instructors were available for one-on-one mentoring. Third, the team wanted to increase student performance and reduce academic attrition by employing student-sensitive instructional techniques.

To achieve these goals, the team implemented a PC-based hybrid instructional system known as the ASTAC intelligent training aid (ITA). The ASTAC ITA has two primary modes:

- Learning Center Mode,
- Instructor Control Mode.

The Learning Center Mode replaced the conventional instructional paradigm with student-sensitive computer-based training. The Instructor Control Mode used serious game technology to replace Device 20F18.

Let us first focus on the Learning Center Mode. This mode employs what we refer to as closed-loop adaptive training technology. The closed-loop adaptive training instructional paradigm uses three building block technologies:

- Adaptive Interactive Multimedia Instruction (IMI),
- Learner Modeling,
- Intelligent Tutoring.

Adaptive IMI rests on the principle that instructional sequences and prescriptions should be tailored to each learner through an individualized training plan (ITP). Essentially, the ITP is continuously assessing which topics to address, in which order, and using which instructional resources. Instructional, practice, and assessment activities are represented as individual instructional objects and dynamically recombined to satisfy the needs of individual learners. With this building-block notion in mind, we can set aside the text-book metaphor that guides the development of conventional IMI and begin to use a tutor as the metaphor. A tutor considers the learning objectives (LOs) that must be addressed as well as what the student already knows. Together, this tells the tutor what he or she must teach the student. Further, the tutor uses what he or she knows about the student to select some instructional activities that are likely to be effective.

After working through the instruction, the tutor builds some assessment activities that are tailored to the student and to the content of instruction. If the student does well, the tutor can move on to new content. If not, then the tutor can select other ways of teaching the material and assessing mastery. This can continue until the student has mastered all of the required LOs.

The adaptive approach allows us to address the needs of divergent populations (like high vs. low achievers or engineers vs. tacticians) and training situations (like initial vs. refresher training or training vs. performance support).

The *InTrain*™ adaptive training engine (McCarthy 2008, McCarthy et al. 1998; McCarthy 1996) was used to support adaptive IMI planning and delivery. *InTrain's* operational sequence is shown in Fig. 13.2. The adaptive IMI engine begins by comparing a user's learner model with the goals set forth in the course data. The comparison results in an ITP that includes what content to deliver and how to deliver it. The planning engine tells the presentation engine what media/activities to present to the student. The presentation engine presents that content and supports user interaction. As the user interacts with the content, the system is constantly monitoring the user's activities and progress toward the specified goals.

As the user's learner model gets updated, the planning engine monitors the training plan and updates it as needed. The preceding discussion emphasized the importance of a learner model to guide adaptive training decisions. The importance of the learner model has been cited not only in the literature related to adaptive IMI, but also in intelligent tutoring systems and web-based intelligent learning environments (Kazi 2004). Recognition of the need to consider the goals, preferences, and knowledge of the user as distinct components of the learner model and to evolve the model based on the user's interaction with the system are ideas frequently expressed in the literature today.

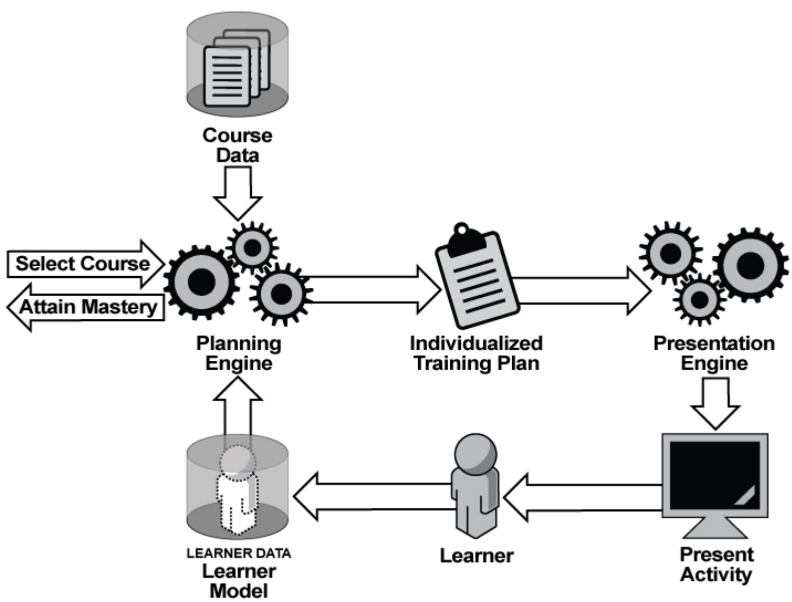


Fig. 13.2 Adaptive IMI operational sequence

In their work, (Wang and Li 2004) stated that domain[1] and user models are important extensions to the learning object paradigm "to support in-depth adaptivity with respect to learners' cognition and to apply technologies developed for intelligent tutoring." They suggested that: Absence of these models prevents Simple Sequencing from "performing adaptivity related to subtle cognitive effects." Research into intelligent agent-assisted adaptivity has included approaches to the dynamic and adaptive configuration of learning plans and to the configuring of the learner model itself; for example, by using a "logic agent" to employ deductive reasoning and infer the needs and intentions of the user within the context of the prescribed learning goals and current learning experience (Baldoni et al. 2002). The proposal for a Mobile Learning Metadata was intended to extend the IMS Global Learning Consortium's Learner Information Package by augmenting the mainly static, domain-independent representation of user traits and characteristics with domain knowledge and learning history information required for more sophisticated learner model adaptivity support (Chan 2004). Mining techniques to acquire student preferences and create user profiles have also been explored, whereby the more typical user "stereotype" established upon first system use is evolved based on the learning path taken and results of tests (Rigou et al. 2004).

The learner modeling approach used in the ASTAC ITA is the Unified Learner Model (ULM). The ULM employs the overlay approach described by (Murray 1991), Endorsement-based Modeling (EBM). One of the key benefits of the Murray approach is that it provides a mechanism to meaningfully combine evaluative information from sources that vary greatly in reliability. Using EBM, the ULM collects evidence for and against the proposition that the student knows a given concept (i.e., a given learning objective (LO)). Evidence may be directly observed (e.g., answers to test questions) or inferred from the objective's place in the hierarchy (e.g., if a parent concept is mastered, there is some evidence that all of its children will be mastered as well). The ULM is primarily an authoritative, managed repository for the LOs within a particular training environment. It also represents a commonly accessible repository of LO mastery information that can be used by multiple training system elements.

The ULM stores "raw" mastery evidence associated with a managed LO for a particular learner rather than a mastery state value. The difference between storing LO mastery evidence and mastery state is to make it possible to determine at some later time what evidence to consider when a mastery determination is made within a particular application context. The mastery evidence items that the ULM stores are referred to as "endorsements" and each endorsement has attributed metadata. Using this approach, it is possible for training system components to select endorsements based on particular attribute values they wish the ULM to consider when making a determination of LO mastery.

[1] "Domain model" as used here refers to an abstract representation of a knowledge domain; for example, in terms of a related collection of concepts, topics, ideas, and facts.

An example would be to discount endorsements that are older than some specific duration (e.g., 6 months) or to include only those endorsements directly attributable to instructor-entered observations. This ability to make determinations of an individual's mastery of specific LOs in real-time, based upon deliberately attributed evidence gathered from a compendium of sources, is how the ULM provides the fuel to power a training system's adaptive decision-making engine.

From time to time, the ITP will call for the student to practice a skill (or to be assessed on skill performance). In these circumstances, an intelligent tutoring system is used to assess performance and, in the case of feedback, provide real-time feedback.

Like the current project, some of the earliest intelligent tutoring systems were developed under military sponsorship (Fletcher 1988). For example, both SOPHIE (Brown et al. 1982) and STEAMER (Hollan et al. 1984) were developed in the early 1980s with funding from the military. SOPHIE used simulation and natural language processing to teach students about troubleshooting electronic systems. STEAMER trained naval officers in the operation of steam propulsion systems. Later in the decade, Massey et al. (1986) described the Maintenance Aid Computer for HAWK – Intelligent Institutional Instructor that was used to train students to maintain the illuminating radar on the U.S. Army's HAWK air defense system. Perhaps one of the most thoroughly researched intelligent tutoring systems in the military domain is SHERLOCK, developed near the beginning of the 1990s (Katz and Lesgold 1993, Lesgold et al. 1992). Like SOPHIE and MACH-III, SHERLOCK was developed to improve troubleshooting skills, this time in conjunction with the F-15 Avionics Test Station. The original SHERLOCK research demonstrated that about 25 hours of practice in that environment had an impact on post-test performance equal to about 4 years of on-the-job experiences (Corbett et al. 1997). Gott et al. (1995) followed up with a report that reviewed five separate evaluations of SHERLOCK. That review showed an overall effect size of about 1.05 standard deviations.

Most intelligent tutoring systems include four components: a Domain Expert, an Instructional Expert, a learner model, and a student-device interface. Together these four components create a student-sensitive "tutor-in-a-box." The Domain Expert is a software module that understands the subject-matter under instruction; it is essentially a synthetic subject-matter expert. Its job is to figure out what should be done in a given situation and then assess the student performance. Just like the Domain Expert knows the subject-matter, the Instructional Expert is a software module that captures our understanding of what it means to be a good coach. The Instructional Expert decides when and how to guide the student's performance. We have already discussed the learner model. In the ASTAC ITA, the adaptive IMI and intelligent tutoring components interact with the same ULM. The student-device interface is a representation of the environment in which the student works. This can range in fidelity from the actual watch station that the student will use, to an immersive VR environment, to a two-dimensional PC representation of the work environment.

The important thing is that the work environment should not get in the way of learning and it should present the user with the same data he/she will encounter in the "real world" while requiring them to make the same decisions.

The ASTAC ITA uses the *ExpertTrain*™ intelligent tutoring engine (see, for example, McCarthy, Pacheco, Banta, Wayne, & Coleman, 1994; McCarthy, Stretton, & Hontz, 1995). *ExpertTrain's* operational sequence is shown in Fig. 13.3. The cycle begins with some stimulus event. That is, something interesting happens in the world and we expect student to react to it. These stimulus events are created through careful consideration of our LOs. Each stimulus event is constructed to stimulate student performance associated with one or more LOs. Once a stimulus event occurs, it must be conveyed to the student. The student-device interface is responsible for this task. After the stimulus event is conveyed to the student, the student must take some action. The Domain Expert's job is to evaluate the appropriateness of the student's actions. At the same time that world events are communicated to the student via the console simulation, the same events are communicated to the Domain Expert. The Domain Expert uses these available data, and only these data, to form a set of expectations for what the student should be doing. The Domain Expert then compares its expectations to the student's actions. This comparison results in an assessment decision.

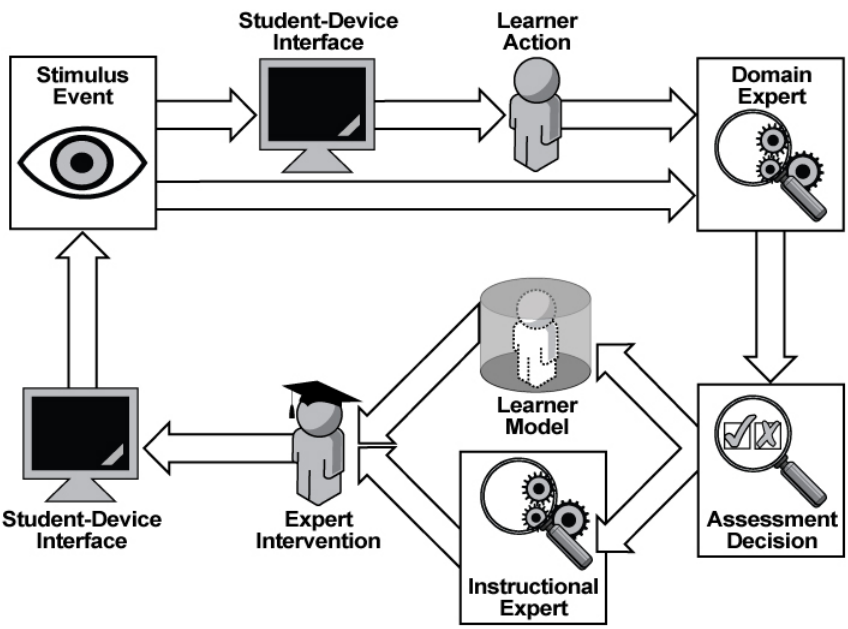


Fig. 13.3 Intelligent tutoring operational sequence

The assessment decision goes to two places. First, the results of the assessment are used to update the learner model. We now know more about the competency of the student and it is important to capture that knowledge. The second place that the assessment decision goes is to the Instructional Expert. Triggered by the assessment decision, the Instructional Expert considers information from the learner model and makes a coaching decision. For example, the Instructional Expert must decide whether to coach at all, which topic to address, how much detail to include, and what modality to use to deliver the coaching. The Instructional Expert's intervention is communicated to the student via the student-device interface and the cycle continues.

More than a dozen tutors have been developed using *ExpertTrain* across a variety of domains. The most comprehensive evaluation of *ExpertTrain* tutoring was conducted by the U.S. Army Research Institute. Wisher and his colleagues compared the training effectiveness of the Multiple Launch Rocket System Virtual Sand Table (MLRS VST) with conventional training approaches. In their study, the MLRS VST demonstrated an effect size of 1.05. That translates to a 35% increase in learning effectiveness (Wisher et al. 2001a, Wisher et al. 2001b).

The other central training mode in the ASTAC ITA is the Instructor Control Mode. The ASTAC ITA Instructor Control Mode complements the Learning Center Mode by providing a venue by which instructors can monitor and assess student performance in a similar manner as previously done when using Device 20F18.

In the Instructor Control Mode, functions are provided to enable the instructor to control platforms (e.g., helicopter, submarines, and ships), and role-play stations as required. Students use the same emulated controls and displays to perform the tasks that they mastered in the Learning Center. However, instead of receiving coaching from the intelligent tutoring system, they receive coaching from the instructor.

Quite simply, the Instructor Control Mode represents a simulation device that supports student and instructor interactions in a suitable operational environment. The ASTAC ITA is supported by the modeling and simulation necessary to support ASTAC/student practice and assessment activities. This includes:

- Shipboard ASTAC operator console emulations,
- Platform simulation,
- Sensor simulation,
- Weapon simulation,
- Environment simulation.

ASTAC ITA simulations are enabled by Sonalysts Naval Simulation Engine™, the core simulator module developed in conjunction with Sonalysts' commercial micro-simulation game, Sonalysts Combat Simulations – Dangerous Waters®.

13.4 Implementation

With these building blocks in mind, let us see how they are combined to form the student experience within the ASTAC ITA. To undertake this review, we will look over the shoulder of Operations Specialist Second Class (OS2) Johnson as she begins to use the Learning Center.

When OS2 Johnson reported to the Fleet ASW Training Center, San Diego to begin her ASTAC training, the Chief Petty Officer in charge of the course, OSC McDaniels, checked her paperwork. Sure enough, OS2 Johnson had completed all her course prerequisites, including passing the ASTAC pre-screening course and completing the pre-screening handbook, and was ready to begin training. OSC McDaniels takes a few minutes to enroll OS2 Johnson in the ASTAC ITA system and assign her to the ASTAC ITA "Initial Training" course. He also takes a few minutes to let OS2 Johnson know what is expected of her and to familiarize her with the structure of the course. When he was done, OS2 Johnson knew that she would have to work hard, but at her own pace, to get through the lessons that she had to master within the time allocated for completion of the course. She also knew that from time to time, she would work one-on-one with OSC McDaniels or another instructor to verify her skills and that those same instructors would be available to her whenever she had questions about the material that she is learning.

When she was ready, OS2 Johnson walked down to the ASTAC learning lab. As she walked into the lab, she noticed the server in the corner and the rest of the PCs arrayed throughout the classroom. Each computer had a large monitor and a headset microphone. OS2 Johnson sat down at her assigned workstation and logged in using the user ID and password that Chief McDaniels had assigned. She knew that her first job was to train the speech recognition system. Chief McDaniels told her that the more she trained the system, the more readily it would understand her. Since she has a bit of an accent, she read several of the training stories and noticed that the accuracy increase. After 45 minutes or so, she decided that the system is working well and that she is ready to get started on the content.

OS2 Johnson clicked on the desktop icon to launch the ASTAC ITA. The login screen appeared and she entered her user name and password. Chief McDaniels had told her that logging in will allow the system to load her personalized "learner model" so that the system can recognize who she is and adapt to her. Her first decision was what mode to access. As a student, OS2 Johnson could either use the Learning Center Mode or the Instructor Control Mode. She was just getting started, so she entered the Learning Center. From there, she picked the "Initial Training" course and navigated down through the lesson hierarchy to the first lesson. She launched the lesson and began training.

The first few lessons were quite easy, focusing on the threat platforms that she must operate against, the aircraft that she will be controlling, and the systems that those aircraft employ. In each lesson, she worked through static (see Fig. 13.4) and interactive media (as it is pictured in Fig. 13.5)

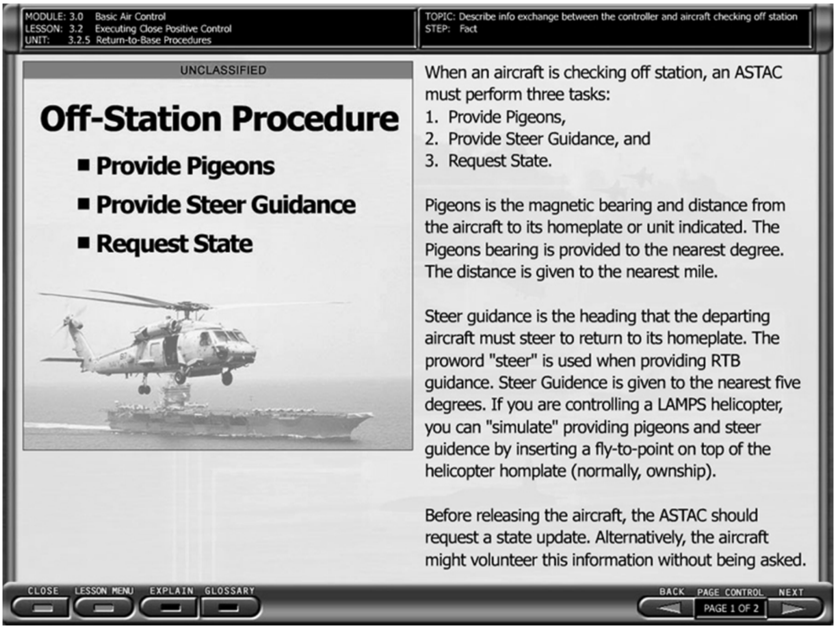


Fig. 13.4 Standard IMI framework

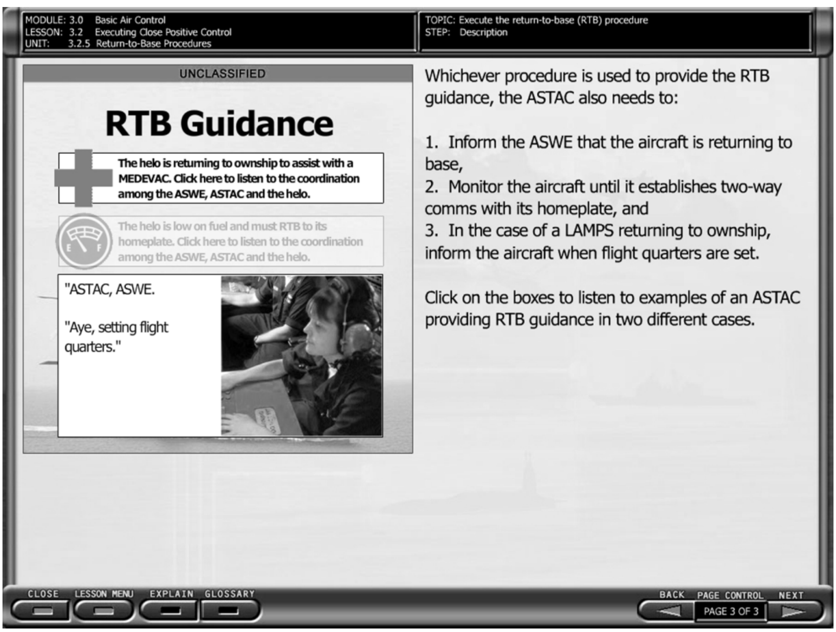


Fig. 13.5 Flash-based IMI framework

The aim is she has to explore the topics, checked her knowledge during practice sessions, and took tests at the end of each lesson. Controls at the bottom of the screen allow her to move back-and-forth through the content, access more detailed instructional prescriptions if she is confused, and display a multimedia glossary that explains key terms and abbreviations.

The Explain button allows the student to view a more detailed explanation of the content, if available. As the name suggests, the Glossary button displays a multimedia dictionary of important terms. The two-pane window at the top of the display allows the student to keep track of their position within the course (and the instructor/mentor to observe student progress over-the-shoulder). The left-hand panel reveals the current lesson tree. The right-hand panel identifies the current learning objective and instructional element. So far, OS2 Johnson was cruising right along and keeping up with most of her classmates.

The first bump in the road came when she had to learn about underwater acoustics. This was all new to her and some of the ideas about how sound behaves as it moves through water took a while to wrap her head around. The practice problems helped. She liked that she got feedback after she answered each question and it only took two or three practice sessions until she could move on to the next LO. When she got to the end of the lesson, she took another test. Unlike her previous tests, this one had some questions that stumped her. She did OK on three of the five LOs, but the system decided that she needed more training on the other two. As she worked through the material a second time, she expected that she would see the same practice items. She was surprised when the items were all different. Even when the same question was used, different answers appeared and they were in a different order. The second pass through the material was much easier than the first, and OS2 Johnson got through the follow-up test with no problems.

About a week into the course, OS2 Johnson was done with the “fundamentals” and was ready to begin controlling aircraft. The first lesson in this area discussed how to accept control of aircraft, including the information that had to be exchanged between the controller and the aircraft, and the procedures that had to be followed. Like the earlier lessons, the instruction here included examples, analogies, and mnemonics to help make the material easier to understand and remember. However, something new popped up in this lesson; when a procedure was introduced; an interactive activity walked OS2 Johnson through the process (see Fig. 13.6). Within these activities, the screen was split into two. Along the right, each step was explained and she was told what to do. The left portion of the screen was devoted to an emulation of her controls and displays. Clicking on the highlighted spot (the circle shown in Fig. 13.6) advanced the demonstration and soon she had a pretty good feel for the procedure. The same way-finding information and navigation controls and appeared at the top and bottom of the display.

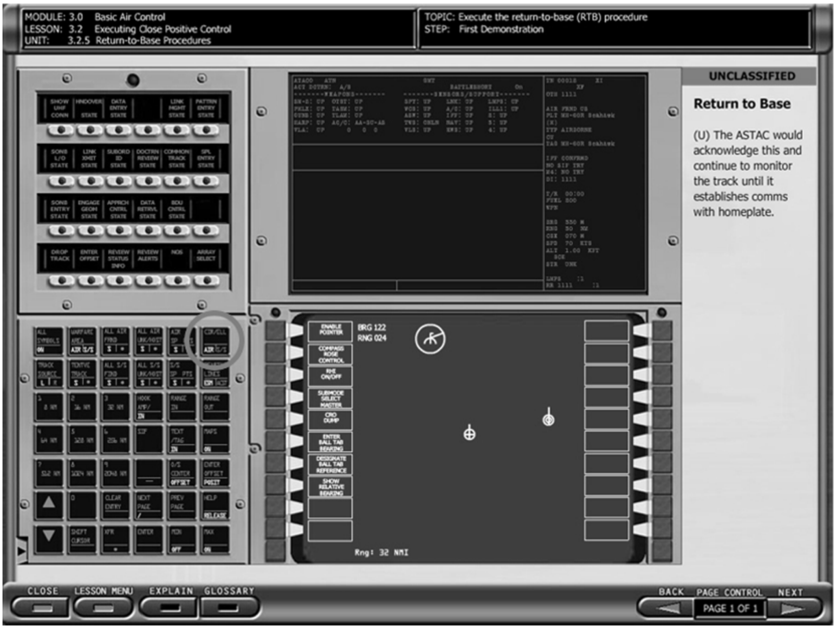


Fig. 13.6 Flash-based IMI demonstration

At least she thought she did. When the system prompted her that she would be entering a simulated environment, she realized that unlike previous lessons that involved just regular test questions, in this lesson she had to really control aircraft. OS2 Johnson remembered that Chief McDaniels had recommended that she periodically update her speech profile, so she left the ASTAC ITA and did some quick training of the voice recognition system by practicing the phrases that she would use in the upcoming exercise. In fact, this training killed two birds with one stone, not only did it help the speech recognition system respond to her better, but also it increased her familiarity with the phrases that would be expected of her. When it was time for practice, the simulation was launched and she was confronted with an array of controls and displays that were reminiscent of those that she had seen on her ship (see Fig. 13.7).

Equipment that was spread out using separate devices on her shipboard console was pulled together on one display. However, they still had the same spatial relationship to each other. For example, the character read-out display that provided textual and numeric information remained above the Basic Display Unit (BDU) that displayed the radar picture.

Hardware buttons that ringed the BDU or that sat off to its left on the console were still there, but now they were graphical user interface controls that worked on the touch screen display. The Variable Action Buttons (VABs) that changed function as OS2 Johnson navigated through the displays still sat upon above the Fixed Entry Keys (FEKs) that remained constant.

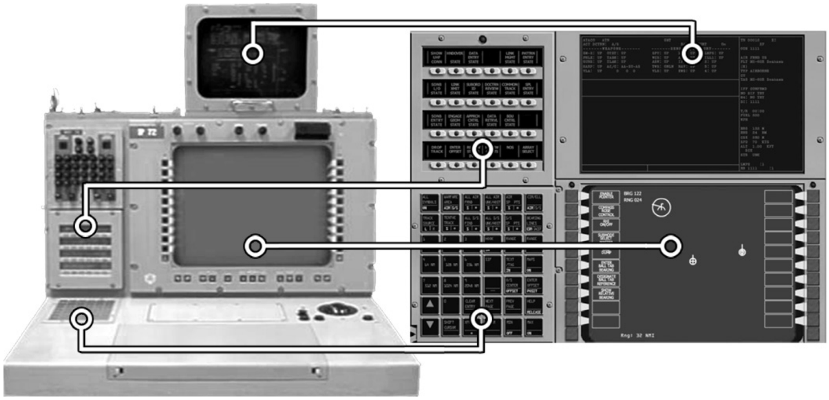


Fig. 13.7 Shipboard console (Left) and software emulation (Right)

At first, some displays did not seem to be there. For example, it took her a second to find the communications panel that normally sat just above the VABs. However, soon she discovered buttons that popped up the communications and other displays (such as a tool that let her monitor and control the ship-to-helicopter communications system).

Although OS2 Johnson knew basically what to do, she was a little clumsy at first. She forgot a step here or there, did one or two procedures the wrong way, and occasionally used the wrong phrase or spoke on the wrong communication circuit. Each time she made a mistake, the coach told her about it, usually with guidance about what she should do instead, and why. The next time she practiced, she got a different scenario and her performance was better. When she did make a mistake, she continued to get feedback and she noticed that it was becoming more and more detailed. After three or four practice scenarios she had the procedure down and was ready to move on.

At the end of the lesson, she took a test. At first, it was just like the other tests. It included the now familiar collection of question types and she was no longer surprised that she seldom saw the same question twice. However, she was surprised that when she was done with the "written exam," she was presented with a "practical exam;" that is, she had to prove her skills in another scenario. As luck would have it, she had trouble with one of the knowledge LOs in the test and got nervous during the aircraft control exercise and made a silly mistake. The system decided that she should take remedial training on those two topics. OS2 Johnson was surprised to discover that, unlike earlier lessons that only had one way to teach the content, this lesson had alternative strategies. On her second pass through the lesson, new, more detailed, instructional objects were used. This new material helped the "light bulb go on" for OS2 Johnson and she cruised through the written exam at the end of the lesson. Further, she was less nervous this time around, maintained her focus, and correctly executed the aircraft control procedure during the practical examination.

OS2 Johnson followed the same basic process on all of the "basic air control" lessons. By the time she was done, she had learned to accept control of an airborne aircraft, launch an aircraft from her ship, execute precision control of an aircraft, and return the aircraft safely to home base. Now she was ready to put her skills to the test in a one-on-one session with Chief McDaniels in the Instructor Control Mode.

The next day, OS2 Johnson and OSC McDaniels arranged to meet in the lab. OSC McDaniels sat down at one of the PCs, slipped on the headset, and logged into the computer. The system recognized him as an instructor based on the permissions assigned by the system administrator. He launched the ASTAC ITA, logged in, and selected the Instructor Control Mode. Once in, OSC McDaniels named the session that he was creating and then selected a network connection protocol. Network connectivity was established quickly and OSC McDaniels was placed in the Instructor Control Mode lobby where he selected a scenario. He reviewed the mission brief that came with the scenario and waited for OS2 Johnson to signal that she was ready to begin the session. Meanwhile, OS2 Johnson had logged into the system, selected the Instructor Control Mode, selected a network connection protocol, selected the session that OSC McDaniels created, and entered the lobby. She reviewed the mission brief for the selected scenario. Because the technology had its start with a commercial game, both OSC McDaniels and OS2 Johnson found it very easy to navigate the system. When she was ready to begin, OS2 Johnson selected the Ready button and her state was reflected on OSC McDaniels' screen.

When he was ready, OSC McDaniels launched the scenario. The ASTAC controls and displays appeared on OS2 Johnson's display and OSC McDaniels saw the instructor control display (see Fig. 13.8). This display let Chief McDaniels see ground truth as well as OS2 Johnson's controls and displays. Using the tools along the bottom of the display, the Chief could control the scenario and take a variety of actions to role-play other members of the team.

The scenario began with the helicopter on deck. OS2 Johnson had to prepare a mission brief for the aircrew. As she gathered the required information, OSC McDaniels monitored her use of the status boards and the accuracy with which she entered the information on the briefing sheet. After the briefing sheets were completed, OS2 Johnson practiced giving the brief to Chief McDaniels.

Further, OS2 Johnson worked through the aircraft launch procedures. At one point, she got stuck and Chief McDaniels prompted her to continue. The rest of the mission went very well. In fact, it went so well that OSC McDaniels decided to "turn up the heat" on OS2 Johnson. First, he created a new aircraft and vectored it in toward the helicopter to see if OS2 Johnson would recognize the danger and alert the crew. When she did, Chief McDaniels directed her to bring the aircraft back home. Along the way, Chief McDaniels reported that a fog bank was rolling in, visibility was reduced and OS2 Johnson had to execute an Emergency Low Visibility Approach. Although the experience stressed her, OS2 Johnson did well.

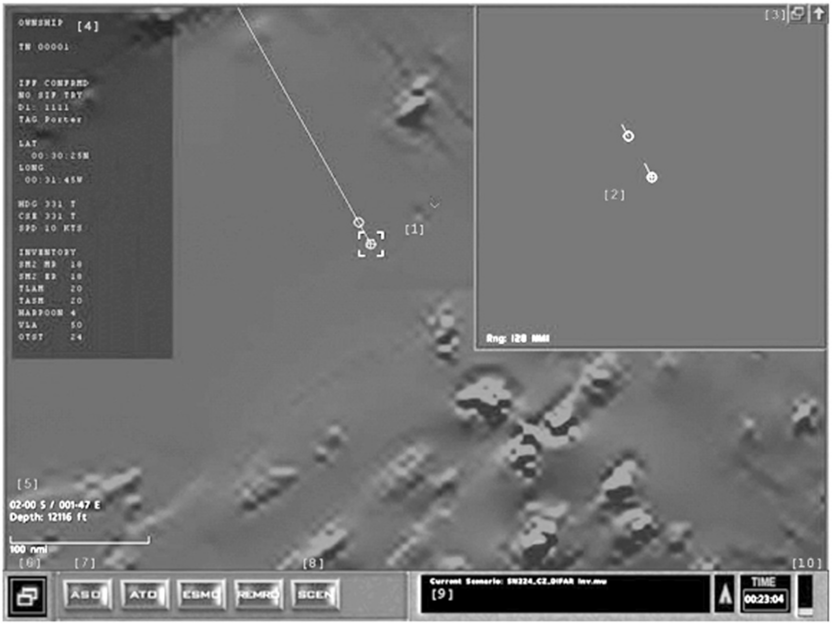


Fig. 13.8 Instructor screen in instructor control mode

After the scenario, OSC McDaniels and OS2 Johnson reviewed her performance using the replay features of the Instructor Control Mode. Together they watched her actions and listened to her communications at critical junctures of the scenario. OSC McDaniels was able to fast-forward to specific scenario times to show OS2 Johnson examples of good performance and to show her times when her performance should have been better. At the end, he gave her permission to move on to the next block of lessons where she began to learn the tactics that she had to apply.

13.5 Results

Recall that the ASTAC ITA was installed at the ASTAC schools to achieve three primary goals. First, the team wanted to replace the aging training Device 20F18 with an easier to maintain PC-based alternative. Second, the team wanted to increase throughput by increasing the amount of time that instructors were available for one-on-one mentoring. Third, the team wanted to increase student performance and reduce academic attrition by employing student-sensitive instructional techniques.

The first goal was achieved in October 2005 when the ASTAC ITA Instructor Control Mode was introduced at the schools on the east and west coasts.

With the introduction of this new training tool, the schools were able to decommission the expensive-to-maintain 20F18 training labs and replace them with more conventional PC-based labs.

The switch to the PC-based emulation of the ASTAC controls and displays was a significant change for the students and instructors. Instead of practicing on outdated displays and controls, the students could now practice using displays and controls that are representative of those that they will actually be using on their ships. The instructors liked the PC controls (point and click, drop down menus, etc.) better than the trainer unique controls resident in the 20F18. In addition, the instructors really liked the ability to create their own Instructor Control Mode scenarios, a capability not provided by the 20F18 simulator.

With the Instructor Control Mode successfully introduced, work then began on developing the Learning Center Mode. The goals of the Learning Center Mode were to increase instructor availability and student throughput. The team's strategy was to introduce adaptive IMI and intelligent tutoring to increase instructional efficiencies and to provide more opportunities for supervised practice.

The Learning Center Mode was accepted by instructors at a slower rate than the Instructor Control Mode. Whereas the Instructor Control Mode replicated the existing training concept of operations with easier to use hardware and software, the Learning Center Mode introduced a new concept of operations. There was a certain reluctance to shift to this new approach. Moreover, the old approach had long historical roots (current instructors had received their initial training using the same curriculum that they were now teaching), the Learning Center Mode was new and was greeted with skepticism. These issues were acerbated when instructors lacked sufficient time to work through the Learning Center curriculum themselves prior to working with students. Thus, when students had questions regarding media-based content or simulation-based tutoring, instructors were initially hard-pressed to answer them and it was easier to attribute it to a failure of the system. These challenges were gradually overcome and the curriculum switched to full adoption of the Learning Center Mode in San Diego in May 2009 and in the Norfolk school in February 2010.

Fig. 13.9 summarizes the data on student drop rate since the introduction of the ASTAC ITA. In Fig. 13.9, we have annotated the point at which the Instructor Control Mode was introduced, the point at which other instructional interventions were introduced, and the point at which the Learning Center Mode was introduced. Note that there are points of inflection that correspond to the introduction of the Instructor Control Mode, the schoolhouse interventions, and the introduction of the Learning Center Mode.

During FY2010, the Learning Center Mode was in use at both training sites. In that fiscal year, the drop rate was 2%, a notable decrease from the preceding years. Historically, the majority of students who cannot complete the course are dropped near the midpoint, after a significant investment has been made in them. In fact, in fiscal year 2010 the cost of each student was approximately $6,500 per week.

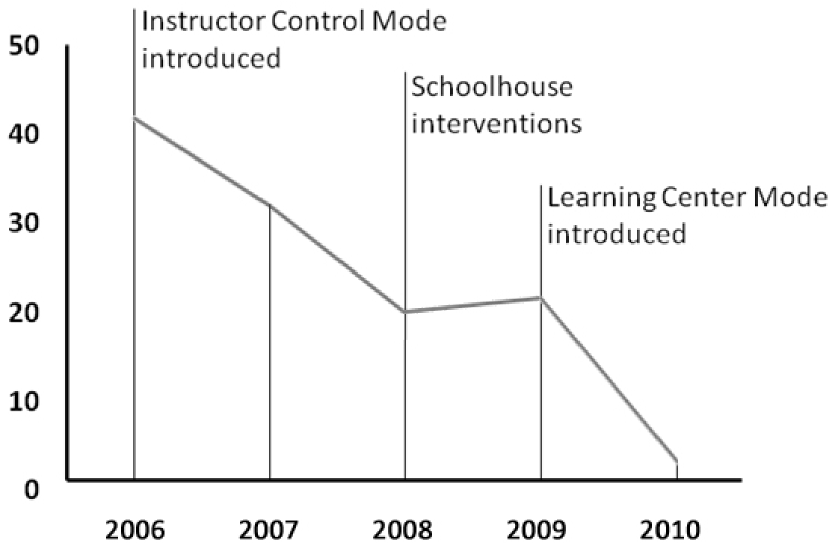


Fig. 13.9 ASTAC drop rate percentage by fiscal year (2006-2010)

This implies that each student who completes the course will cost approximately $62,400. Therefore, if the student drops at the midpoint of the course, an investment of approximately $31,200 has been lost. If we assume a drop rate of 28% (as was reported for FY2000-2005) produces a total of 34 drops in a given year, this represents a loss of $1,060,800 every year. Decreasing this drop rate to 2% would produce a yearly savings of nearly a million dollars.

13.6 Discussion

The ASTAC COI employed a legacy training system. In addition to being expensive to maintain, the aging device severely limited the practice opportunities that were available to students. This lack of practice opportunities directly contributed to an unacceptably high drop rate. To address these concerns, the Center for Surface Combat Systems worked with Sonalysts to replace the legacy system with the ASTAC ITA. The ASTAC ITA included an Instructor Control Mode which allowed instructors and students to work together as they had historically. It also included a Learning Center Mode that used adaptive IMI and intelligent tutoring technology to create and maintain an individualized training plan for each student. The Instructor Control Mode provided an immediate return on investment while maintaining the historical training concept of operations. The Learning Center Mode introduced a very different concept of operations that was more student-centered and self-paced.

Drop rates from the first fiscal year in which it was employed indicate that the Learning Center may also be making a notable contribution to reduced attrition.

The experience of introducing the Learning Center Mode has emphasized the importance of preparing technology users for technology acceptance. During the development of the ASTAC ITA, several steps were taken to assure this preparedness. Most notably, the instructional staff was an integral part of the development team. ASTAC instructors advised the development team throughout the project and reviewed each instructional artifact as it was produced.

However, several factors limited the value of the integration. First, it is important to note that the ASTAC School located in San Diego is the “lead” school for the ASTAC COI and that Norfolk has a subordinate role. However, since Norfolk was geographically closer to the development team, that school had a greater leadership role early in the development effort. San Diego participation increased toward the end of the project, but technology acceptance may have been greater if they had participated more fully from the beginning.

Second, because instructors rotated into and out of the project as their assignments changed, it was difficult to maintain a consistent perspective on the project. Some instructors served as better advocates for the effort than others and it was difficult to maintain a consistent picture of the content to be taught. In a related vein, instructors from San Diego and Norfolk sometimes had different perspectives on the instructional content and it was not uncommon to require meetings to resolve these conflicts. It was not always possible to achieve consensus at these meetings and one party or the other would leave not fully approving of the curriculum.

Third, not all instructors were able to participate in the development process. For these instructors, the ASTAC ITA was something foreign that was being thrust upon them. This was especially problematic for the ASTAC ITA Learning Center Mode. Although there was a two-week period in which the technology was introduced to instructors at both schools, not all instructors had an opportunity to participate. Moreover, the two-week period was not sufficient to allow all instructors to complete the revised COI. In retrospect, this is critical. It is vital that every instructor who is to mentor the course work through the entire course prior to working with students. This familiarity is critical to prepare the instructors to answer student questions.

Finally, it is important to note the significant change in the instructor role with the introduction of the ASTAC ITA Learning Center Mode. With this transition, instructors went from the primary providers of information (i.e., “a sage on the stage”) to mentors that had to complement content provided by an external agency (i.e., “a guide on the side”). In large part, the training that Navy instructors received did not prepare them for this change and greater on-site support during the transition period would have made this transition significantly easier.

13.7 Conclusions

Since the introduction of the ASTAC ITA, the underlying technology has continued to evolve. The most significant advances have been to the Instructor Control Mode. Recently, the U.S. Air Force has adopted the basic Instructor Control methodology and applied it to the training of satellite operators. Under this project, a number of advancements have been possible. For example, the original Instructor Control Mode allowed an instructor to assess the performance of a single student, the current version supports simultaneous assessment of multiple students. To support this expansion, the current version of the Instructor Control Mode includes some low-level automatic assessment of operator performance. This assessment function verifies that the student has satisfied various performance milestones in a timely fashion. If a student falls behind, then an icon changes alert the instructor to the problem. Drill-down functions allow the instructor to assess the nature of the student's error and to provide the required coaching.

The current version of the Instructor Control Mode also allows students to participate in distributed simulations. This feature allows them to participate in multi-disciplinary exercises with teammates located elsewhere while their performance is being assessed in real-time by instructors.

Finally, a software development toolkit was developed that allows new simulations and operator consoles to be developed within the basic Instructor Control Mode architecture. This toolkit will allow a range of developers to produce training systems for various satellite systems without having to replicate the basic training functions built into the underlying architecture.

The second aspect of the ASTAC ITA that has continued to evolve is the ULM. The ULM is at the heart of the adaptation process used by the ASTAC ITA; it is the system's representation for each individual student is, what he or she knows, and what instructional material he or she has experienced. Each student has his or her own ULM record and it is updated in real-time as the student interacts with the system. Originally, the adaptive IMI and intelligent tutoring environments used separate learner models. In the course of ASTAC ITA development, these representations were merged to form the ULM. Since the release of the ASTAC ITA, ULM development has continued. For example, the ULM was rehosted as a web service consistent with the Sharable Content Object Reference Model (SCORM) 2004 (Advanced Distributed Learning 2006). This service allowed SCORM-conformant content objects to record multi-dimensional mastery evidence to support adaptive instructional decisions and detailed performance reports (McCarthy and Scroggins 2010). In conjunction with this work, a ULM Toolkit with a number of learner model management features was created to allow for the registration of client applications, the registration of users, and the creation and management of LOs.

The development team is considering future enhancements. One enhancement is to the processing of "practice." Currently, when a practice-to-success plan is used, students practice until they have demonstrated satisfactory performance.

Experience has indicated that it would be beneficial to allow students to "break out" of practice to receive remedial training on the LO in question.

Similarly, we are investigating using error patterns within the intelligent tutoring environment to diagnose whether performance deficiencies are indicative of a lack of declarative knowledge or procedural skill. Often, by the time that students enter the intelligent tutoring environment, the errors they make can be attributed to a lack of fluency. This condition can be recognized through a decreasing error frequency across the training opportunity. If the total error count decreases over some performance window, it is likely that fluency is at fault for the remaining errors and that the best remediation strategy is additional practice.

However, there may be cases in which the cause of errors should be diagnosed as a declarative knowledge flaw, rather than a lack of procedural fluency. Two error patterns can be used to diagnose this type of flaw. The first error pattern reflects a lack of knowledge of the procedure being considered. If over some window of attempts the error count remains steady or increases, the chances of a declarative flaw (rather than a fluency flaw) increase. If the errors involve a significant proportion of all the steps in the procedure, it is likely that the student is fundamentally unknowledgeable about the procedure and that a complete retraining on the procedure would be useful. On the other hand, if the errors are concentrated on a few steps, the declarative flaw is likely to be more localized. If the errors are consistent (i.e., of the same bug class), then it is likely that the student holds a mistaken belief. However, if the errors are random (i.e., occur across bug classes), then it is likely that the student simply is unknowledgeable with respect to these steps.

The second error pattern reflects a lack of knowledge of how to diagnose or respond to a particular procedure. This type of error reflects a mismatch between the demands of a situation and the student response. This flaw could result if the student misdiagnosed the situation or had a faulty mapping between the situation and the appropriate response. This is recognized when the student correctly applies the wrong solution to a given situation.

The third enhancement that we are investigating is rehosting the ASTAC ITA as a web-accessible application. While the hybrid instructional strategy embodied in the ASTAC ITA is proving to be very successful, it is limited to those installations that have both resident instructors and the local area network (LAN)-based ASTAC ITA software (namely, schoolhouse/Fleet training installations at San Diego, CA; Norfolk, VA; Mayport, FL; Everett, WA; Pearl Harbor, HI; and Yokosuka, Japan). The development team is currently considering extending the applicability of the ASTAC ITA by transitioning it from a LAN-based architecture to a web-based one. This transition will allow a much broader population to benefit from use of the ASTAC ITA training system.

For example, a student on a ship should be able to take an ASTAC ITA lesson on the ship via the Learning Center Mode, then work with a real instructor off ship at a remote location via the Instructor Control Mode.

In summary, the introduction of the ASTAC ITA and other interventions at the schoolhouse resulted in a substantial reduction in student drop rates. The ability of the ASTAC ITA to deliver a consistent and repeatable training standard ensures all students are trained to a common standard and removes the inherent subjectivity of Instructor-led curriculum.

Acknowledgments. InTrain and ExpertTrain are trademarks of Sonalysts, Inc. All other trademarks are the property of their respective owners.

References

Baldoni, M., Baroglio, C., Patti, V.: Supporting users in adaptive web-based applications: Techniques from reasoning about actions. In: Proceedings of WOA, pp. 49–55 (2002)

Brown, J.S., Burton, R.R., de Kleer, J.: Pedagogical, natural language and knowledge engineering techniques in Sophie I, II and III. In: Sleeman, D., Brown, J.S. (eds.) Intelligent Tutoring Systems. Academic Press, London (1982)

Corbett, A.T., Koedinger, K.R., Anderson, J.R.: Intelligent tutoring systems. In: Helander, M.G., Landauer, T.K., Prabhu, P.V. (eds.) Handbook of Human-Computer Interaction. Elsevier Science B. V., The Netherlands (1997)

Chan, T., Sharples, M., Vavoula, G., Lonsdale, P.: Educational metadata for mobile learning. In: Proceedings of WMTE, p. 197 (2004)

Fletcher, J.D.: Intelligent training systems in the military. In: Andriole, S.J., Hopple, G.W. (eds.) Defense Applications of Artificial Intelligence: Progress and Prospects, Lexington Books, Lexington (1988)

Gott, S.P., Kane, R.S., Lesgold, A.: Tutoring for transfer of technical competence. Technical report, Air Force: AL/HR-TP-1995-0002 (1995)

Hollan, J.D., Hutchins, E.L., Weitzman, L.: STEAMER: An interactive inspectable simulation-based training system. Artificial Intelligence Magazine 5, 15–27 (1984)

Katz, S., Lesgold, A.: The Role of the tutor in computer-based collaborative learning situations. In: Lajoie, S.P., Derry, S.J. (eds.) Computers as Cognitive Tools. Erlbaum, Hillsdale (1993)

Kazi, S.A.: A conceptual framework for web-based intelligent learning environments using SCORM-2004. In: Proceedings of ICALT, pp. 12–15 (2004)

Lesgold, A., Lajoie, S., Bunzo, M., Eggan, G.: SHERLOCK: A coached practice environment for an electronics troubleshooting job. In: Larkin, J., Chabay, R. (eds.) Computer-Assisted Instruction and Intelligent Tutoring Systems: Shared Goals and Complementary Approaches. Lawrence Erlbaum Associates, Hillsdale (1992)

Massey, L.D., Kerland, L., Tenney, Y., de Bruin, J., Quayle, K.C.: HAWK MACH-III intelligent maintenance tutor design development report. Technical report BBN, Cambridge, MA: BBN Laboratories, 6315 (1986)

McCarthy, J.: Military applications of adaptive training technology. In: Lytras, M., Gaševiće, D., Ordóñez de Pablos, P., Huang, W. (eds.) Technology Enhanced Learning: Best Practices. IGI Publishing, Hershey (2008)

McCarthy, J.E.: The InTrain™ system: Past and future. Talk Presented to the Conference on Cognitive Approaches to Work Behavior, Pennsylvania State University, PA (1996)

McCarthy, J.E., Johnston, J., Paris, C.: Toward development of a tactical decision making under stress integrated trainer. In: Proceedings of I/ITSEC (1998)

McCarthy, J.E., Pacheco, S., Banta, H.G., Wayne, J.L., Coleman, D.S.: The radar system controller intelligent training aid. In: Proceedings of I/ITSEC (1994)

McCarthy, J.E., Scroggins, R.S.: Developing a SCORM-conformant learner model. IEEE Learning Technology Newsletter 12(3), 38–40 (2010)

McCarthy, J.E., Stretton, M.L., Hontz, E.B.: A classroom evaluation of an intelligent training aid. In: Proceedings of HFES (1995)

Murray, W.R.: An endorsement-based approach to student modeling for planner-controlled intelligent tutoring systems. Technical paper. Air Force Systems Command, Brooks Air Force Base, Texas AL-TP-1991-0030 (1991)

Rigou, M., Sirmakessis, S., Tzimas, G.: An architecture for an adaptive web-based learning environment. In: Proceedings of WBE (2004)

Wang, H.-C., Li, T.-Y.: Considering model-based adaptivity for learning objects. Learning Technology Newsletter 6(2), 9–11 (2004)

Wisher, R.A., Macpherson, D.H., Abramson, L.J., Thornton, D.M., Dees, J.J.: The virtual sandtable: Intelligent tutoring for field artillery training. Research report, U.S. Army Research Institute for the Behavioral and Social Sciences, 1748 (2001a)

Wisher, R.A., Banta, H.G., Macpherson, D.H., Abramson, L.J., Thornton, D.M., Dees, J.J.: The adaptation and evaluation of an intelligent tutoring system for the multiple launch rocket system. In: Proceedings of I/ITSEC (2001b)

Abbreviations

ASTAC	ASW/ASUW Tactical Air Controller
ASUW	Antisurface Warfare
ASW	Anti-Submarine Warfare
BDU	Basic Display Unit
COI	Course of Instruction
EBM	Endorsement-based Modeling
IMI	Interactive Multimedia Instruction
ITA	Intelligent Training Aid
ITP	Individualized Training Plan
LO	Learning Objective
MLRS VST	Multiple Launch Rocket System Virtual Sand Table
OS2	Operations Specialist Second Class
PC	Personal Computer
ULM	Unified Learner Model

Chapter 14
CELTS: A Cognitive Tutoring Agent with Human-Like Learning Capabilities and Emotions

Usef Faghihi[1], Philippe Fournier-Viger[2], and Roger Nkambou[3]

[1] University of Memphis
Tennessee, USA
usef.faghihi@gmail.com
[2] University of Moncton
Moncton , Canada
philippe.fv@gmail.com
[3] University of Quebec at Montreal
Montreal, Canada
nkambou.roger@uqam.ca

Abstract. To provide a rich learning experience, an intelligent tutoring agent should be able to take into account past and present events, and to learn from its interactions with learners to continuously improve the assistance it provides. Until now, the learning capabilities of tutoring agents in educational technologies have been generally very limited. In this chapter, we address this challenge with Conscious-Emotional-Learning Tutoring System (CELTS), a cognitive tutoring agent, whose architecture is inspired by the latest neuroscientific theories. CELTS implements human-like learning capabilities such as episodic, emotional, procedural and causal learning. The agent is used in a simulation-based tutoring system for learning the complex task of operating the Canadarm2 robotic arm on the International Space Station (ISS). Experimental evaluations showed that the agent's learning capabilities considerably enhance its adaptation to learners during interactions and consequently improves learners' performance.

14.1 Introduction

Professional human tutors are capable of taking into account past and present events and are driven by social concerns. To support optimal learning, a virtual tutoring agent should have similar capabilities. Given that training human learners involves dynamic interactions between the tutor and learners, a virtual tutoring agent must adapt itself to the learner's needs and profile. It should also be able to learn from its interactions with learners.

A. Peña-Ayala (Ed.): Intelligent and Adaptive ELS, SIST 17, pp. 339–365.
springerlink.com

Consequently, to evaluate learners, the ideal tutoring agent should possess learning capabilities similar to those of human beings including emotional, episodic, and causal learning. Such capabilities improve its decision-making process during interactions and thus allow for a better adaptation to the learner. Current intelligent agents' learning capabilities are limited to the use of only one type of learning or a few types in a single learning mechanism (Vernon et al. 2007). However, it has been shown in neuroscience that the various types of learning are functionally incompatible with one another (Sherry and Schacter 1987). It is therefore a great challenge and an important research problem to conceive an intelligent tutoring agent that integrates several types of human-like learning mechanisms and uses them to enhance its tutoring assistance to learners. It is worth mentioning that the learning mechanisms discussed in this paper are implemented in a distributed manner in our CELTS (Dubois et al. 2007, Faghihi et al. 2011).

CELTS is a complex cognitive tutoring agent inspired by the latest neuroscientific findings. It integrates three important learning mechanisms:

1. The first mechanism is *Emotional Learning* (EML). It simulates human emotions based on neuroscientific evidences and allows CELTS to learn desirable and undesirable situations when interacting with the learner, thus influencing the decisions made by CELTS and its adaptability;
2. The second mechanism is *Episodic Learning* (EPL). It simulates the multiple-trace theory of memory consolidation. It makes the agent constantly learn from previous interactions with learners. EPL allows CELTS to remember learner's mistakes and proposes the best solution according to his or her profile;
3. The third mechanism is *Causal Mechanism* (CLM). It allows CELTS to learn the causes of events, and then to make predictions about future events in order to make the best pedagogical decisions, given the most likely causes. It is implemented by using sequential pattern mining and association rules algorithms.

Using these three mechanisms, CELTS is capable of automatically constructing and updating its emotional, episodic, and causal knowledge to provide better and more tailored assistance to learners.

The rest of this chapter is organized as follows. Sect. 14.2 discusses the problem of building adaptive educational systems and the problem of implementing human-like learning in cognitive agents. Sect. 14.3 presents CELTS's architecture and explains how the three aforementioned learning mechanisms were integrated into it. Sect. 14.4 reports how CELTS has been successfully applied in a tutoring system for teaching the highly-difficult task of operating Canadarm2, the robotic arm installed on the ISS. An empirical evaluation of CELTS shows that the learning mechanisms greatly improved CELTS's capabilities of adaptation to learners. The experiments also show that CELTS equipped with different types of learning capabilities improved learners' performance. Finally, the paper ends in Sect. 14.5 with a conclusion and a brief indication of future works.

14.2 Related Works

Not all educational systems have the same capacity to adapt to learners. For instance, some educational systems offer no adaptation because they are “hard-coded.” Some of the most advanced works on adaptive educational systems have been done in the Artificial Intelligence in Education community (AIED) (Woolf 2009). These works aimed at developing “Intelligent Tutoring Systems” (ITS), which are capable of providing tailored assistance to learners without human interventions (Woolf 2009).

In an ITS, the key to generating tailored assistance to learners is the “student model” (VanLehn et al. 2005, Woolf 2009) . It represents student competencies, learning achievements, and the learner’s knowledge (VanLehn 1988). However**,** the student model can also contain other types of information about the learner such as his/her learning preferences. During a training session, ITS regularly updates its beliefs about the learner (VanLehn 1988, Woolf 2009) and adapts its behavior accordingly. For instance, ANDES is an ITS that uses a student model implemented with a Bayesian network to keep track of the student’s performance and generate tailored assistance (VanLehn et al. 2005). Furthermore, adaptive hypermedia systems (Brusilovsky and Millan 2007) usually rely on student models to generate tailored assistance for learners in web-based learning environments.

So far, most works on the adaptation capabilities of educational systems have focused on how to build and exploit a student model. In this chapter, we discuss another aspect that has an important impact on the adaptation capabilities of an educational system. It aims at extending an educational system with learning capabilities that enable new behaviors built from its knowledge according to its interaction with learners. This will help both the educational system and learners to improve their performance. We believe this research is complementary to the research on student modeling.

Several works aiming at building intelligent agents with human-like learning mechanisms have been done. Cognitive architectures such as SOAR, ACT-R, and CLARION (Anderson et al. 2004, Nason and Laird 2005, Sun 2001, Sun 2006) are the most popular results of these research efforts. However, all these fail to implement solutions that are cognitively plausible. In fact, they are limited to using only one type of learning or a few types in a single learning mechanism.

For instance, the SOAR architecture can only learn new procedural knowledge called production rules (Vernon et al. 2007). Sun also claimed that in the CLARION architecture, the explicit (declarative)/implicit (non-declarative) knowledge’s interact in a synergetic way to solve a problem and to learn a specific task. However, in the current version of CLARION, during bottom-up learning the propositions (premises and actions) are already present in top level (explicit) modules before the learning process starts, and only the links between these nodes emerge from the implicit level (rules). Thus, there is no unsupervised causal learning for the new rules created in CLARION (Hélie 2007).

In the ACT-R architecture, a few learning mechanisms have been integrated for procedural learning, but it does not address other important types of learning such as emotional learning and causal learning. Franklin also claimed that the Learning Intelligent Distribution Agent (LIDA) is equipped with different types of learning (Franklin and Patterson 2006). However, no experiment has been conducted about LIDA learning mechanisms. In the next section, we introduce our cognitive agent CELTS, and then describe how we integrated several learning mechanisms into its architecture.

14.3 The Conscious Emotional Learning Tutoring System

CELTS is a hybrid intelligent tutoring cognitive architecture integrated in CanadarmTutor (Nkambou et al. 2006), a simulation-based ITS for learning how to operate the Canadarm2 robotic arm installed on the ISS. The CanadarmTutor learning environment is a 3D reproduction of Canadarm2 on the space station and its control panel (Fig.14.1 A). Learning activities in CanadarmTutor mainly consist of operating Canadarm2 for performing various real-life tasks with the simulator such as carrying loads with the robotic arm or inspecting the ISS.

Operating Canadarm2 is a complex task because astronauts have to follow a strict security protocol. Furthermore, the arm has seven degrees of freedom (seven joints that can be rotated), and users only have a partial view of the environment through the cameras that they choose and adjust. CanadarmTutor was the subject of several research projects (Nkambou et al. 2006).

In this chapter, we present CELTS, the component of CanadarmTutor that acts as the core tutor that takes all the pedagogical decisions, generates dialogue and performs the high-level assessment of the learner. CELTS is a cognitive agent architecture based on Baars' theory (1997) of consciousness. It is constructed with simple agents called “codelets” (which reproduce Baars' “simple processors”). The central point of the system is the “access consciousness,” which allows all resources to access centrally selected information that is “broadcast” to unconscious processes (which guides the agent to be stimulated only with the most relevant information). CELTS performs through cognitive cycles. A cognitive cycle in CELTS starts by perception and usually ends by the execution of an action. CELTS uses its Behavior Network (BN) for action selection (Fig.14.1.B). The BN is implemented based on Maes' Behavior Net (Maes 1989). It is a network of partial plans that analyses the context to decide what to do and which type of behavior to set off (Fig.14.1.B).

Given that CELTS acts as a tutor, an expert can define different solutions in the BN to help learners. Thus, BN's nodes are pedagogical actions such as scaffolding messages, hints, demonstrations, etc. (Fig.14.1.D) to assist learners while they manipulate Canadarm2 in the virtual environment (Fig.14.1.A).

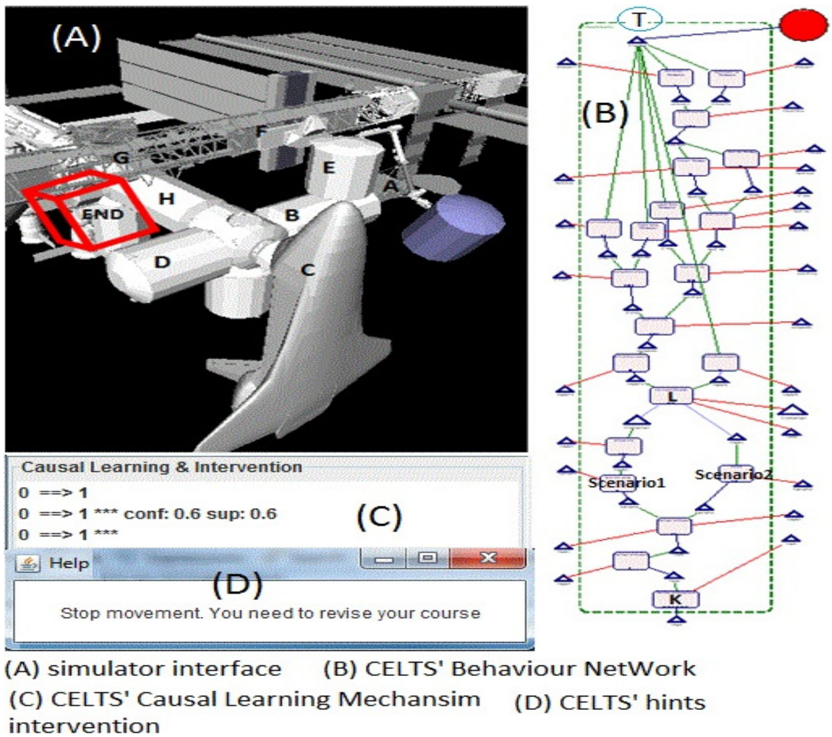


Fig. 14.1 CELTS interface and Canadarm2

The learners' manipulations of the robotic arm in the virtual environment constitute the information perceived by CELTS. Those perceptions go through CELTS cognitive cycles and the process will end by an action (e.g., sends the learner advice to improve their current performance). Our team has now added different types of learning into CELTS (Faghihi et al. 2011). For the rest of this section, we briefly explain how emotional, episodic, and causal learning are implemented in CELTS.

14.3.1 Emotional Learning

Emotions influence cognition, and vice versa (Damasio 2000, LeDoux 2000, 2006). To integrate a generic computational model of emotion in CELTS, we have been inspired by research in neurosciences. Following Ledoux (2000), we take it that the amygdala subserves an additional memory system, which we call emotional memory.

But the amygdala's involvement in learning and memory goes beyond emotional memory, as it modulates learning in other memory systems, especially declarative memory (Schoenbaum et al. 2000). Squire and Kandel (2000) explain in the next excerpt: "The amygdala and the hippocampus systems independently support non-declarative and declarative memories. The two systems can work together. Animals retain a task more strongly, when a variety of hormones such as adrenaline are injected into their blood and brain after they learn to perform a task. The enhancement of memory by emotion results from the amygdala's influence on declarative memory (pp. 171-172). Other experiences show that the more active the amygdala is at the time of learning, the more it enhances the storage of those declarative memories that had emotional content." (p.173)

Accordingly, we describe two general types of emotional learning: *pure emotional learning* (i.e., learning subserved by the amygdala), which gives rise to emotional memory proper, and *emotionally modulated learning* (i.e., learning subserved by hippocampus and cortex but that is modulated by the amygdala), which brings about other types of memories and infuses them with emotional content.

Each of these types of emotional learning corresponds to a specific pathway to the amygdala. The first route, the short route, is based on peripheralistic concepts from James' work (James 1884).

It is short and direct; information flows from the sensory thalamus directly to the amygdala (ESR rectangles in Fig.14.2) and then projects to particular structures such as the basal ganglia. The short route enables implicit (i.e., unconscious) direct behavioral reactions based on previous rewards or punishments associated with the same or similar stimulus (Rolls 2000, Squire and Kandel 2000).

Human reactions are then rapid and unconscious (Squire and Kandel 2000) because the reaction is dependent on information that is not processed by other brain structures, notably cortical structures. For example, if, while walking in a forest, we encounter a long and sinuous cylinder-like object close to our leg, we will in general react very quickly and, without thinking, move our leg away from the object. In this case, information from the retina entered the sensory thalamus, which passed the information along to appropriate cortical structures for further analysis.

But the signal was also sent to the amygdala, which recognized the possible danger posed by the perceived object and sent a signal to the motor system for immediate movement of the leg, away from the object.

In the second route, based on centralistic concepts originating from Cannon's work (Cannon 1927), (ELR rectangles in Fig.14.2), information from the external environment is analyzed by various cortical areas (primary sensory cortex, unimodal associative cortex, polymodal associative cortex). It is then sent to the hippocampus for memory retrieval and temporary storage.

All this processing serves to interpret the external stimuli, to give it meaning (categorization by the cortex) and link it to other events in episodic memory (see below), before it goes to the amygdala for emotional appraisal and response.

In this example, longer route corresponds to the recognition that the object we moved our leg away from is not a snake after all but a peculiarly twisted piece of wood, and the remembrance of previous forest walks in which we saw branches.

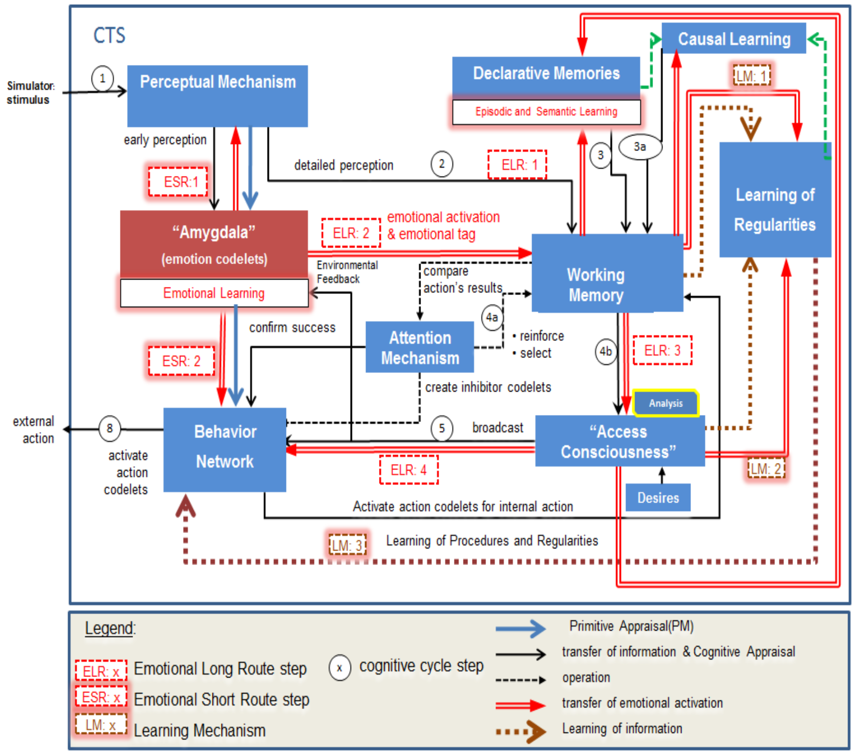


Fig. 14.2 CELTS' architecture with emotion and learning mechanisms

Although it is slower, the response produced by this second route possesses the normal phenomenology of thoughtful behavior and can be consciously controlled. Once it has been interpreted by cortical structures, the information then flows back to the amygdala where it can serve to reinforce or correct its initial processing of the information. Next we explain how emotions are implemented in CELTS.

14.3.1.1 Emotions in CELTS

For CELTS, the two routes of emotions are important. We illustrate this with an example. Consider that an astronaut is manipulating Canadarm2 in the virtual world and that information coming from the simulator to CELTS describes an imminent collision. A collision is a very dangerous situation on the ISS and CELTS must immobilize the arm immediately. Thus, CELTS sends an urgent message to ask the astronaut to stop Canadarm2. This can be said to be a quick reaction produced by the short route based on an emotion of imminent threat. Once CELTS has sent the message, CELTS interprets the information received from the virtual world more attentively and engages in a dialogue with the learner.

CELTS asks questions such as: “Does the manipulation of Canadarm2 really cause a danger? What is the distance between Canadarm2 and ISS?” These reactions can be seen as being influenced by the long route of emotions.

In CELTS cognitive cycles, when a percept enters Working Memory (WM) (see Fig.14.2) as a single network of codelets, emotional codelets which are situated in CELTS “pseudo-amygdala”, the Emotional Mechanism (EM), inspect each coalition, and infuse it with a level of activation proportional to its emotional valence[1]. This increases the likelihood that the coalition draws attention to itself. This emotional intervention on the coalitions in WM is how CELTS EM gets involved in CELTS long route (ELR rectangles in Fig.14.2).

Attention influences the EM by providing information about the environment regarding the discrepancy between what was expected and what effectively occurred. This may alter the future valence assigned by EM to situations in the environment, as well as the importance EM gives to a situation. In our model, after each interaction with the environment, CELTS EM updates its information (especially in dangerous situations) about its surrounding environment for future situations. Thus, the importance of any given situation may increase or decrease in CELTS next encounters with it.

CELTS can make two reactions when faced with a dangerous situation. We now explain how the information, coming from CELTS Perceptual Mechanism, flows along the short route and the long route (ESR and ELR in Fig.14.2). The *short route* (see ESR rectangles in Fig.14.2) starts with perception just like the long route (see ELR rectangles in Fig.14.2). The perception codelets connect in parallel both to CELTS Behavior Network (BN) (Fig.14.1.B) and to its emotional codelets. The activation is sent directly by perception codelets to emotional codelets is the first stage of the short route. The EM establishes the positive or negative emotional valence[1] of the event for the system. The valence assigned to the event may result from evolution (an innate valence accorded to evolutionarily important situations) or from learning.

Thus in CELTS, some emotional codelets might correspond to innate (designed) sensitivities (e.g., to excessive speed for Canadarm2, or to an imminent collision). Other emotional codelets may have learned the valence of situations from experience. Either way, emotional codelets possess direct connections to behavior nodes in the BN (Fig.14.1.B), to which they send positive or negative activations.

Some of these emotional codelets react more strongly than others and so send out stronger valence activations to the behavior nodes. If the valence activations exceed a behavior node’s firing threshold, then the corresponding action will fire automatically. This emotional intervention reflects a direct route between the amygdala and bodily responses, influencing action selection. This corresponds to James’ theory (James 1884) about why a bodily reaction generates an emotional feeling, if an important stimulus directly causes the bodily reaction.

---

[1] Emotional valences represented as values between -1 and +1.

Whichever route was responsible, short or long, the firing of a behavior node generates one or more expectation codelets. These are a type of Attention codelets[2]. These codelets are processes that watch for the arrival in WM of a given piece of information, expecting to see, within a given time frame, some specific result(s) for the action taken by CELTS. The expectation codelets have a duty in CELTS. They serve as "environmental reinforcers" to the Action Selection Mechanism in the BN. If they see information coming in WM that confirms the Behavior's expected result, they directly send reinforcement activation to the behavior nodes that created them (that is, they do not do so through conscious broadcasting). This behavior will thus see its base-level activation heightened, making it a more likely choice in a similar context.

In the case of a failure to meet expected results, however, relevant resources need to be recruited, to allow them to analyze the cause of the failure, to correct the previous emotional interpretation of the situation, and to allow deliberation to take place concerning supplementary and/or alternative actions.

The expectation codelets then work to have discrepancies brought to the attention of the whole system (in an eventual conscious broadcast of the noted discrepancy) by sending the information to the CELTS WM. After sending the information to WM, CELTS continues through its cognitive cycles (see next subsection, step two to eight of the cognitive cycle) to allow for improved decisions.

The expectation codelets' second duty concerns our "pseudo-amygdala", the EM, in cases where it forced an automatic reaction through the short route (e.g., the imminent collision in the virtual world).

Indeed, when low-level basic information coming from the perception codelets recognizes aspects of the situation as highly dangerous, there is no time to think and, through the mechanism described above, the emotional codelets will force an action to fire in the Behavior Network. This makes CELTS jump before thinking (James 1884) (ESR path, red-dotted rectangles, and blue arrows which demonstrate primitive appraisal in Fig.14.2). That is, it makes CELTS act before it has time to become "conscious" of the situation and consciously plan a course of action. This corresponds to the first reaction taken by CELTS in our aforementioned example about imminent collisions in the virtual world.

However, the instantaneous, mindless reflex must be evaluated following the more thorough analysis of the situation that comes later, through the *long route*. CELTS can do this because the short and long routes process the information in parallel. In fact, instinctive reactions execute faster. Eventually, however, a conscious broadcast of information (see next subsection, Step 5 of CELTS's cognitive cycle), which gives CELTS a better idea of the situation, allows normal action selection to take place.

---

[2] The task of attention codelets is to find their own specific content in the WM and send them for the consciousness competition. For example, one codelet may look for a node representing fear. When an attention codelet finds its content, it creates a coalition containing this content and related content. The coalition, then, is sent for consciousness.

When the action thereby proposed comes into WM (see next subsection, Step 2 of the cognitive cycle), the expectation codelets compare it to the reflex action that has been prompted. If roughly in correspondence, they put into WM a confirmation to the effect that the initial reaction was right, which will serve, when broadcasted, as a reinforcer to the emotional codelet(s) that were instrumental in setting off the reflex. In effect, this will make our pseudo-amygdala reinforce the relevant rules and nodes. However, when the initial reaction diverges from the behavior proposed by the more detailed analysis, the pseudo-amygdala has to alter its first reaction. This corresponds to Step two in our example about imminent collisions.

From a neurological point of view, control over actions is the role of cortical areas. In CELTS, the expectation codelet that determined the action taken by the short route was inappropriate subtracts some activation from the codelets in the EM responsible for the implicit reaction. Activation will also be subtracted from the corresponding nodes in the BN that executed the action.

This way of control seems in accordance with the fact that the amygdala never unlearns a "rule," especially for very dangerous stimuli, and always reacts to a given stimulus (Rolls 2000, Squire and Kandel 2000). This description highlights the fact that CELTS EM, which responds implicitly to events, reacts faster than the conscious process, but may react in ways that are different from what conscious planning would decide. Emotional codelets receive reinforcements from the environment (via expectation codelets) and can learn or create new nodes for the actions they took. In the next section, we explain how the EM influences CELTS cognitive cycle.

14.3.1.2 Impact of Emotions in CELTS Cognitive Cycle

The emotional long route involves the consciousness mechanism. Emotions influence this mechanism at every step in the cognitive cycle. We briefly recall each step in the cycle and then, in italics, explain how the valence attributed to situations by CELTS EM influences it. For a visual representation of the described process, please refer to Fig.14.2.

Step 1: The first *stage* of the cognitive cycle is to perceive the environment, that is, to recognize and interpret the stimulus.

Step 2: The percept enters WM. The percept is brought into WM as a network of information codelets that covers the many aspects of the situation. In this step, if the received information is considered important or dangerous by the EM, there will be a direct reaction from EM which primes an automatic behavior from BN (Purves et al. 2008, Rolls 2000, Squire and Kandel 2000).

Step 3: Memories are probed and unconscious resources contribute. These resources react to the last few consciousness broadcasts (internal processing take more than one single cognitive cycle). What is brought back from episodic memory is evaluated by the emotional codelets (Fig.14.2) and receives its emotional load anew.

Step 4: Coalitions assemble. In the reasoning phase, coalitions of information are formed or enriched. Attention codelets join specific coalitions and help them compete with other coalitions toward entering "consciousness." Emotional codelets observe WM content, trying to detect and instill energy to codelets that they "believe" require it, and attach a corresponding emotional valence. As a result, emotions influence which information comes to consciousness and modulate what will be explicitly memorized.

Step 5: The selected coalition is broadcast. The Attention mechanism spots the most energetic coalition in WM and submits it to the access consciousness, which broadcasts it to the whole system. With this broadcast, any subsystem (appropriate module or team of codelets) that recognizes the information may react to it.

Steps 6 and 7: Here unconscious behavioral resources (action selection) are recruited. Among the modules that react to broadcasts is the BN. BN plans actions and, by an emergent selection process, decides upon the most appropriate act to adopt. The selected behavior then sends away the behavior codelets linked to it. In this step, the emotion codelets stimulate nodes in the BN, preparing it to react, priming certain behavior streams, and thereby increasing the likeliness of their firing. This mostly mimics priming effects. The emotional valence (positive or negative) attached to the published coalition will influence how resources react. When the BN starts a deliberation for action, for instance to build a plan, the plan is emotionally evaluated as it is built, the emotional codelets playing a role in the selection of the steps in the plan. If the looping (through the cognitive cycle) concerns the evaluation of a hypothesis, the emotional codelets give it an emotional evaluation, perhaps from learned lessons from past experiences.

Step 8: Action execution. Motor codelets stimulate the appropriate muscles or internal processes. Emotions influence the execution, for instance in the speed and the amplitude of the movements.

14.3.1.3 How CELTS's Emotional Mechanism Learns

CELTS is equipped with implicit and explicit emotional learning. *Implicit emotional learning* occurs when EM nodes reaction intensity or the strength of its connections to nodes in WM or BN (Fig.14.1.B) is modified. In the implicit emotional learning phase, the influence of emotional codelets (either those temporary resident in WM or those situated in EM and listening to the received information) through their base level activation affects the creation of coalitions and their selection (Step 5 of cognitive cycle) by the Attention Mechanism (Steps 3 and 4 of cognitive cycle). The ELM, in its implicit learning phase, learns which coalition in WM received emotional energy from EM.

This occurs when emotional codelets resident in WM try to detect which coalition, according to the agent's goal, is emotionally more important than others. The emotional codelets then attach themselves to those coalitions, and they instil a portion of emotional energy in them.

This may increase the likelihood of the emotionally selected coalition drawing Attention to itself in the upcoming cognitive cycles. Moreover, ELM learns that it must send energy to these emotional codelets in WM to prolong the coalition's lifetime in WM and to help them be selected by AM. This is because codelets with no energy will exit WM. Thus, the emotional codelets detected as emotionally important by EM will remain active in WM to attach themselves to coalitions. This emotionally-learned information will never be forgotten by the system (Squire and Kandel 2000, Westen 1999).

Explicit emotional learning occurs following the broadcasting of information (Step 5 of cognitive cycle) in the system. In the explicit emotional learning phase, if for a given situation, EM detects no emotionally important material in the information coming to WM (considered as very important by perceptual nodes in Step 4 of cognitive cycle), it will create a new, empty node with a context which describes ongoing events. To fill out the action part of the new node, EM will wait for the consciously-mediated selection of behavior and the ensuing broadcasting of the event with external confirmation after the execution of the action by CELTS. If the selected action from BN received a strong (positive or negative) reinforcement from the environment, EM learns the broadcasted information instantaneously, that is, in less than a second. Note that CELTS processes information through cognitive cycles, which happen five times per second (Franklin and Patterson 2006).

At this point, EM has associated the context of the new node with the action selected and executed by CELTS. Information brought to consciousness right after the action took place becomes the result part of the created node. In CanadarmTutor, CELTS is equipped with EM learning for two reasons. First, the integration of EM in CELTS allows CELTS to deal with dangerous situations such as when the learner performs a collision risk, a collision or bad manipulations of Canadarm2.

In these situations, the EM makes CELTS learn to react quickly to prevent the learner from performing dangerous mistakes (in CanadarmTutor, a dangerous mistake is for example hitting the space station with the robotic arm, which could cause severe damages in real-life to the space station and the arm).

In the next paragraphs, we present such situations where the EM had to intervene explicitly in CanadarmTutor. The second reason why we integrate the EM in CELTS is that it is used for other learning mechanisms such as the episodic learning mechanism that will be described in Section 14.3.2.

A first example of a situation where the EM intervenes is when the user is asked to move Canadarm2 from configuration *A* to configuration *B* on the ISS. CELTS must recognize which movements will not cause collisions. CELTS then gives the user feedback in the form of questions or hints. In the following we explain the execution of CELTS with EM and without EM.

For this case, consider the execution without EM. When the learner brings Canadarm2 close to ISS, the simulator informs CELTS there is a risk of collision. The data is then selected by CELTS Attention mechanism and broadcast to the system. After deliberation from CELTS BN, an action is chosen and shown to the user (e.g, Do you know what the distance is between Canadarm2 and ISS?).

To react to this situation, CELTS uses the long route. No significant changes are made to the energy in the BN. Now, consider the execution with the EM. When the learner brings Canadarm2 too close to ISS, the simulator immediately informs CELTS that there is a risk of an imminent collision, and this collision risk is considered very dangerous. As a result, EM codelets react to the situation by instilling enough negative energy (equal to -0.9, a very negative valence) to the corresponding behavior in the BN to make it fire. The BN reacts to the situation by prompting the message to the user: "Stop moving the arm"; "This is a very dangerous situation"; "Answer the following questions before moving on". Because this situation, it is attributed a high emotional valence (high-threat situation), CELTS short route is activated.

In parallel, CELTS long route also activates. As a result of the high emotional valence, the collision risk information received from the virtual world is more attentively examined. CELTS then asks the user the following question: "Do you know what the distance is between Canadarm2 and ISS?" If the user answers correctly, the emotional codelets' intensity decreases. The second question is "If you get closer to ISS, what will happen?" Again, if the user selects the correct answer, the emotional codelets' intensity converges to a positive value. This means the user is an expert. Accordingly, the intensity of the emotional codelets that reacted to the collision risk must very rapidly reach a positive value. On the contrary, if the user fails to answer, CELTS considers the user to be a beginner.

The intensity of the emotional codelet that reacted to this event reaches -1, the most negative value. At this stage, the user will be prompted not to perform any further movement and will review the lesson. The emotional intensity will remain at -1 if the user does not stop manipulating the Canadarm2. If the user stops manipulating Canadarm2, the negative emotional intensity will reach zero after a number of additional cognitive cycles.

We now describe a second situation where the EM has to intervene. It is when the learner performs camera adjustments. Camera adjustments are one of the most important aspects of Canadarm2 manipulation. At any moment, the learner has to select the best three cameras (from a set of about twelve cameras on ISS) for viewing the environment (since no camera offers a global view of the environment). Of course, forgetting camera adjustment is not as dangerous as collision risk. However, forgetting camera adjustment may lead users to manipulate Canadarm2 very close to ISS, which in turn increases the risk of a collision with ISS.

The execution of situation two without EM is similar to situation one. Thus, we only explain here the execution of situation two with EM. In this situation, consider that the initial emotional valence is zero. After a while, WM receives information indicating the user has forgotten to adjust the cameras. Given that the information does not suggest a very dangerous situation but it is nonetheless important, EM attributes a -0.5 emotional valence to it. In effect, it is important enough for CELTS to select it and bring it to consciousness (long route). After deliberation, a hint reminds the user to perform camera adjustment (e.g., Did you forget to do something?).

At this stage, EM codelets react indirectly to the situation. EM codelets reaction depends on the outcome of the user-CELTS interaction. If CELTS questions (e.g., What else did you forget?) are correctly answered, the intensity of EM codelets for direct reaction will decrease. However, if the user does not answer CELTS questions correctly, the codelets' intensity increases. This negative valence increase will occur during every user-CELTS interaction or during any bad Canadarm2 manipulation. When the user finally understands the problem and adjusts the cameras, the EM codelets negative energies will decrease. If the user does not stop moving Canadarm2, EM short route is activated, thus reacting directly to the situation, as explained in situation one. CELTS will react to the collision risks in the same manner as detailed in situation one.

We remind readers that situation one and two are mentioned as examples to illustrate the role of the EM in CELTS. However, the role of the EM is not limited to these two situations. In fact, what is presented above as emotional learning and emotional interventions applies to all interactions between the learner and CELTS. During interactions with learners, the EM assigns positive or negative emotional valences to each interaction between CELTS and learners. This helps CELTS better construct its different memories such as episodic, causal and procedural memories. In the next section, we explain how emotions help the construction of CELTS Episodic Memory.

14.3.2 Episodic Learning

The second learning mechanism that is integrated in CELTS is the Episodic Mechanism (EPM). It simulates the multiple-trace theory of memory consolidation (Faghihi et al. 2010). The multiple-trace theory postulates that every time an event causes memory reactivation, a new trace for the activated memory is created in the hippocampus.

Memory consolidation occurs through the reoccurring loops of episodic memory traces in the hippocampus and the construction of semantic memory traces in the cortex. We have implemented the EPM in CELTS as three processes: 1) recording interactions with learners in separate memories; 2) performing memory consolidation on these recorded memories to generate an episodic memory; 3) using the episodic memory to adapt CELTS behavior.

14.3.2.1 Recording Interactions with Learners

The first process consists of recording all the information broadcast in the system during a training session between CELTS. In our context, CELTS learns during astronauts' training sessions for arm manipulation in the Canadarm2 simulator virtual world. A trace of what occurred in the system is recorded in CELTS's different memories during consciousness broadcasts (Faghihi et al. 2010).

In our implementation, each sequence of interactions during a training session is recorded as a sequence of events in a sequence database. Each event in CELTS, denoted as *X=(ti, Ai),* represents what happened during a cognitive cycle. The timestamp *ti* of an event indicates the cognitive cycle number, whereas the set of items *Ai* of an event contains an item that represents the coalition of information (e.g., collision risk with ISS) that was broadcast during the cognitive cycle.

For example, Table 14.1 shows an example of a database produced by user manipulations of Canadarm2 in the virtual world containing six short sequences. Consider the first sequence. The first event of sequence *S1* indicates that during cognitive cycle 0, due to arm manipulation by the learner, coalition *c1* was broadcasted and an emotional valence of -0.8 for emotion *e1* (high threat) was associated with the broadcast. The second event of *S1* indicates that at cognitive cycle 1, coalition *c2* was broadcasted with emotional valence -0.3 for emotion *e2* (medium fear) and that behavior b1 was executed.

If the event *S1* appears several times during learners' interactions with CELTS, the following pattern can be discovered: forgetting to adjust the cameras or incorrectly adjusting their parameters is followed by a collision risk.

14.3.2.2 Memory Consolidation

The second process of CELTS EPL is memory consolidation. It extracts frequently occurring events from past experiences to construct the Episodic Memory (Faghihi et al. 2010). In our implementation, this is achieved by a sequential pattern mining algorithm. As an example, Table 14.1 depicts a database respecting the extracting patterns that appear in at least two sequences; and Table 14.2 shows some sequences obtained from the application of the algorithm on such a database.

Table 14.1 A data set of 6 sequences

ID	Events sequences
S1	<(0, c1 e1{-0.8}), (1, c2 e2{-0.3} b1), (2, c4 b5)>
S2	<(0, c1 e1{-0.8}), (1, c3), (2, c4 b4), (3, c5 b3)>
S3	<(0, c2 e2{-0.3}), (1, c3), (2, c4), (3, c5 b3)>
S4	<(0, c3), (1, c1 e1{-0.6} b4),(2, c3)>
S5	<(0, c4 b4), (1, c5), (2, c6)>
S6	<(1, c1 e1{-0.6} b4), (2, c4 b4), (3, c5)>

Table 14.2 Some patterns found

Mined sequences	Support
<(0, c1 e1{-0.7}), (2, c4)>	66 %
<(0, c3), (2, c5 b3)>	33 %
<(0, c4 b4), (1, c5)>	50 %
<(1, c3), (2, c4), (3, c5 b3)>	33 %

Thus, the first frequent pattern is < (0, *c1 e1* {-0.7}), (2, *c4*)>, which was found in sequences *S1*, *S2*, *S4,* and *S6*. Because the events containing *e1* in these sequences have numeric values -0.8, -0.8, -0.6 and -0.6, the algorithm calculated the average when extracting that pattern, which resulted in the first event having e1 with value {-0.7}. Note that in CELTS, the memory consolidation process is performed periodically to constantly keep the episodic memory up to date.

14.3.2.3 Using the Episodic Memory to Adapt CELTS Behavior

The third process of the CELTS EPLM is to use the episodic memory to adapt CELTS behavior to learners. The idea is to modify CELTS so that it uses its Episodic Memory to make better decisions. Concretely, CELTS does this by constantly checking if the current situation matches with patterns from previous situations in the Episodic Memory (Faghihi et al. 2009). If some pattern matches, CELTS then chooses to follow the pattern that had the most positive emotions in CELTS (positive emotions means the learner corrected his/her mistakes or performed well).

We now present an example to illustrate how CELTS EPL can use the aforementioned extracted information to better help learners. This example is taken from experiments performed with users during training sessions with *Canadarm-Tutor*. The example addresses the situation of a collision risk caused by a bad camera adjustment. For moving Canadarm2, it is a fact that users must first perform camera adjustments.

In our experiments, we noted that users frequently forgot this step, and moreover, users frequently did not realize they had neglected this step. This increases the risk of collisions in the virtual world. We thus decided to implement this situation as a *medium-threat* situation in CELTS BN (see Fig. 14.3.B). For this situation, experts defined different scenarios in CELTS BN.

When CELTS is faced with this situation, it has to choose between three possible actions to help the learner (named “scenario 1”, “scenario 2”, and “scenario 3” on Fig.14.3.B). These actions are to: 1) give a direct solution such as “You must stop the arm immediately”; 2) give a brief hint such as “I think this movement may cause some problems. Am I wrong or right?”; 3) give a proposition such as “Stop moving the arm and revise your lessons.”

To choose between these three possibilities, CELTS relies on the EPL mechanism to make the best decision according to what has been successful or not in the same situation previously. We next describe the three scenarios and explain how CELTS chose between them for a given user at a given moment.

The first scenario is to give the solution to the learner. CELTS first evaluated this scenario. CELTS EPL detected that CELTS EM attributed *negative* valences to this scenario in the past. The following pattern is an example of the sequences extracted by the data mining algorithm: <(t = 0, *c1*), (t = 1, *c2*),(t = 2, *c3*),(t = 3, *c4*),(t = 4, *c5*), *e*{-0.9}>.

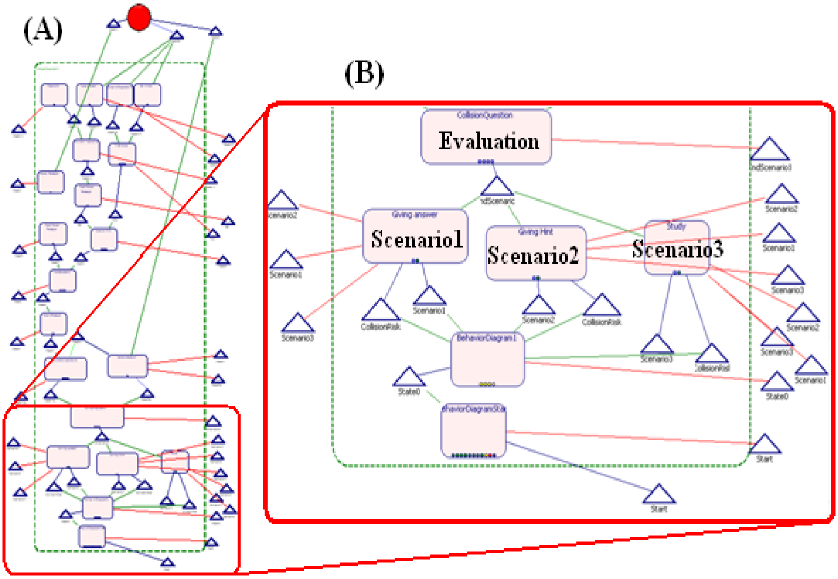


Fig. 14.3 Part of the CELTS' Behavior Network

The mean emotional valence for this sequence; the emotional valences given by EM to each event in the sequence are not shown. The sequence contains the following data: at time 0, the broadcast coalition c1 indicates a collision risk was imminent. At time 1, the broadcasted coalition *c2* indicates that CELTS gave the answer to the user. At time 2, the broadcasted coalition *c3* indicates the user did not know why there was an imminent collision risk. At time 3, the broadcasted coalition c4 indicates CELTS gave a hint to the user. At time 4, the broadcasted coalition c5 indicates scenario1 received an emotional valence equal to -0.9 from CELTS' EM due to the user's answers. In this scenario, users received direct solutions from CELTS, but nonetheless failed to react properly. This failure thus led CELTS EM to associate the negative valence -0.9 to the emotion e1 (medium fear). Because negative emotions are associated to this scenario, CELTS concluded that this scenario is not a good action to take for the collision risk problem.

The second scenario is to give a hint to the learner. CELTS then evaluated this scenario. CELTS EPL detected that CELTS EM attributed *positive* valences to this scenario. The following sequence is an example of pattern extracted by the data mining algorithm: <(*t*=14, *c2*), (*t*=15, *c21*), (*t*=16, *c22*), (*t*=17, *c23*), *e2* {0.7}>.

Again, the mean emotional valence for this sequence, the emotional valences given by EM to each event (in each step) in the sequence are not shown. The scenario 2 contains the following information: at time 14, the broadcasted coalition *c2* indicates a collision risk was imminent in the virtual world. At time 15, the broadcasted coalition *c21* indicates *give a hint*. At time 16, broadcasted coalition *c22* indicates *give the answer*. At time 17, broadcasted coalition *c23* indicates the user's reaction was correct.

As a result, CELTS EM attributed a positive emotional valence of 0.7 to the sequence as a whole. Given these positive emotional valences, CELTS concluded that giving a hint is a good action if the user forgetting to perform camera adjustments (Fig.14.3, scenario 2).

The third scenario makes a suggestion to the learner. CELTS then evaluated this scenario. CELTS EPL detected that CELTS EM attributed zero emotional valence to this scenario. The next sequence is an example of those extracted by the data mining algorithm: < (t=44, c2), (t=45, c51), (t=46, c52), (t=47, c53), e {0} >. It is noted that we here show the mean emotional valence for this sequence. The emotional valences given by EM to each event (in each step) in the sequence are not shown. Scenario 3 contains the following information: at time 44, the broadcasted coalition c2 indicates a collision risk was imminent in the virtual. At time 45, the broadcasted coalition c51 indicates the following message "Please revise your course". At time 46, the broadcasted coalition c52 indicates user is inactive. At time 47, the broadcasted coalition c53 indicates the user decided to stop software. As a result, CELTS's EM has attributed a zero emotional valence to the sequence as a whole. Given the zero emotional valence, CELTS concluded that this scenario is not appreciated by most users. This scenario is not a good candidate for the collision risk problem. Therefore, finally, CELTS applied scenario 2.

Episodic learning in CELTS seeks the sequences with the most positive emotional valences and highest frequencies. In our example, the event (*t* =14, c2), (*t* =15, c21) met these requirements. In future cases, if the emotional valence is not as positive as was the case in our example, CELTS may choose another scenario rather than scenario2. Because the set of patterns is regenerated after each CELTS execution, some patterns emerge, while others disappear, depending on the sequences of events stored by CELTS.

This ensures that CELTS behavior changes over time in the case that some scenarios become more or less negative. It also ensures that CELTS can adapt its behavior to different learners. The above episodic learning allows CELTS to adapt better to users by remembering users' mistakes from previous training sessions and reusing solutions that received positive feedback from learners (carrying positive emotions). However, using only sequential pattern mining algorithms, CELTS was not capable of finding the cause of the learners' mistakes. In the next subsection, we explain how we equipped CELTS with a causal learning mechanism to find the cause of the learners' mistakes.

14.3.3 Causal Learning

The third learning mechanism we have integrated in CELTS is CLM. CLM allows CELTS to learn the causes of learners' mistakes. To our knowledge, two research groups have attempted to incorporate CLM in cognitive agents. The first is Schoppek with the ACT-R architecture (Anderson 1993), who has not included a role for emotions in this causal learning and retrieval processes.

ACT-R constructs the majority of its information according to the I/O knowledge base method. It also uses a sub-symbolic form of knowledge to produce associations between events.

As explained by Schoppek (2002), in ACT-R, sub-symbolic knowledge applies its influence through activation processes that are inaccessible to production rules. However, the causal model created by Schoppek in ACT-R "overestimates discrimination between old and new states." The second is Sun (2006), who proposed the CLARION architecture. In CLARION current version, during bottom-up learning, the propositions (premises and actions) are already present in top level (explicit) modules before the learning process starts, and only the links between these nodes emerge from the implicit level (rules). Thus, there is no unsupervised causal learning for the new rules created in CLARION (Hélie 2007).

Various causal learning models have been proposed, such as Gopnik's model (2004). All proposed models use a Bayesian approach for the construction of knowledge. Bayesian networks work with hidden and non-hidden data and learn with little data. However, Bayesian learning needs experts to assign predefined values to variables (Braun et al. 2003). Another problem for Bayesian learning, crucial in the present context, is the risk for combinatory explosion in the case of large amount of data. In our case, constant interaction with learners creates the large amount of data stored in CELTS module.

For this last reason, we believe that using data mining algorithms is more appropriate. In particular, we chose a sequential rule mining algorithm to implement the causal learning mechanism in CELTS. The advantage of causal learning using the sequential rule mining algorithm is that CELTS can learn in a real-time incremental manner – that is, the system can update its information by interacting with various users. A final reason for basing CELTS causal learning on a data mining algorithm is that the aforementioned problem explained by Schoppek, which occurs with ACT-R, cannot occur when using sequential rules for causal learning.

However, although data mining algorithms learn faster than Bayesian networks when all data is available, they have problems with hidden data. Furthermore, like Bayesian learning, there is a need for experts, since the rules found by data mining algorithms must be verified by a domain expert (Braun et al. 2003).

In the next two subsections, we describe in detail CELTS CLM and put forward its advantages and limits. We first describe how the causal memory is created and then how it is used by CELTS to adapt its behavior.

14.3.3.1 Causal Learning in CELTS

In CELTS the causal memory is created by a sequential rule mining algorithm, where sequential rules are association rules respecting the temporal ordering of events (Faghihi et al. 2011). Causal knowledge is generated in CELTS in three steps: 1) the information is broadcast in CELTS; 2) a decision is made by CELTS about the ongoing problem; 3) a decision is reinforced by next experiences while CELTS receives data from the world and interacts with learners (Fig.14.3.A.D).

So using EML, EPL, CLM, CELTS memorizes learners' errors and finds the causes of the errors by means of extracting rules from the sequences of events recorded by the EPL. These sequences of events are the interactions that took place between CELTS and users during Canadarm2 use by users in the virtual world.

As explained in Sect. 14.3.2, they are saved to different CELTS memories respecting the temporal ordering of the events that occurred between users and CELTS. The retrieved sequences of events contain nodes (Fig.14.1.B). Each node contains at least an event and an occurrence time.

For instance, consider the situation illustrated in Fig.14.1.B.D; different interactions may occur between users and CELTS depending on whether the nodes' preconditions in the BN (Fig.14.1.B) become true. To find the causes of the problem produced by the users in the virtual world, we chose RuleGrowth, a data mining algorithm that we developed in previous work (Fournier-viger et al. 2011).

The algorithm extracts sequential rules (e.g., X → Y , where X and Y are unordered sets of events) between sets of events with their confidence and support[3] (Agrawal et al. 1993) from all past events. That is, if X occurs, Y is likely to follow. The original RuleGrowth algorithm takes a database of event sequences (as previously defined) and two thresholds as input: a minimum support and a minimum confidence threshold. It outputs the set of rules of the form X→Y that have a support and confidence greater or equal to these thresholds, where $X \cap Y = \emptyset$.

This information can be interpreted as an estimate of the conditional probability $P(Y \mid X)$ (Deogun and Jiang 2005, Hipp et al. 2002, Li and Deogun 2005). By extracting rules as explained above, respecting the temporal ordering of events, we have demonstrated CELTS is capable of inductive reasoning (Faghihi et al. 2010).

However, one problem that occurs when applying RuleGrowth into CELTS is that there can be up to several thousands of rules that are found when the sequence database is large. At any given moment, only a few rules are generally relevant.

If too many rules are found, it degrades the performance of the data mining algorithm and also of CELTS, which has too many rules to consider for reasoning. To reduce the number of rules and to extract the most precise and relevant ones, we adapted the RuleGrowth algorithm to add constraints on events that a rule contains so that the rules that are relevant to CELTS are extracted in any situation.

To explain how we performed this modification, we first describe how RuleGrowth proceeds to find rules. RuleGrowth first counts the support of each individual event in a database and then creates all rules between two single events. The algorithm then discovers all larger rules by recursively adding one event at a time to rules already found either to the left part or the right part. RuleGrowth integrates various strategies to perform this search efficiently, while minimizing the number of database scans that is required (Fournier-viger et al. 2011).

[3] Given a transaction database D, defined as a set of transactions $T=\{t1,t2\ldots tn\}$ and a set of items $I=\{i1, i2,\ldots in\}$, where $t1,t2,\ldots,tn \subseteq I$, the support of an itemset $X \subseteq I$ for a database is denoted as sup(X) and is calculated as the number of transactions that contain X. The support of a rule X→Y is defined as $sup(X \rightarrow Y) / |T|$, where $sup(X \rightarrow Y)$ is the number of transaction where X occurs before Y. The confidence of a rule is defined as $conf(X \rightarrow Y) = sup(X \rightarrow Y) / sup(X)$.

We now explain how the four constraints we have defined are used to discover rules that are relevant for CELTS. The constraints are shown in the following:

- C1: the set of events that the left part of a rule can contain,
- C2: the set of events that the right part of a rule can contain,
- C3: the set of events that the left part of a rule has to contain,
- C4: the set of events that the right part of a rule has to contain.

The constraints can be used to discover rules that answer the following four questions:

Q1: What may happen? Constraint C1 is used to force rules to contain only events that occurred in the left part of the rule. For instance, if the event {Forget-camera adjustment} and {Selected wrong arm joint} occurs, CELTS can put these two events in C1 and discover rules such as forgetting camera adjustment and selecting wrong joint will causes collision risk, which indicates this combination of events could result in a collision risk. CELTS can then use this information to stop the learner or take any other remedial action.

Q2: What was the cause of the event? Constraint C4 is used to force rules to contain events that occurred in the right part of the rule. This allows CELTS to create explanations of why these events occurred. For instance, in one situation, CELTS recorded that Collision risk (CR) occurred. By searching the part of the rules containing constraint C4, CELTS was able to find that forgetting camera adjustments and choosing the wrong joint are the most likely causes of the problem.

Q3: Given a specific situation, which events will take place? Using constraint C1 and/or C3, we make sure events will be registered in the left part of the rules, and by using constraints C2 and/or C4 we make sure that the prediction will be registered in the right part of the rules. By using this strategy, CELTS can obtain rules that predict the occurrence of some specific events in the future. For instance, if a learner moves the Canadarm2 too close to the ISS, CELTS can put this event in constraint C3 and the collision event in constraint C4, to know if it is likely that a collision with ISS will occurs. This information can then be used to take the appropriate action.

Q4: What may happen after given action of CELTS? Given a situation, constraint C1 is assigned to all registered events that have happened and constraint C3 is assigned to all registered action(s) that CELTS can take for the situation. By using these constraints, CELTS can discover rules that indicate the consequences of the possible actions to be taken. For instance, if while manipulating Canadarm2, the learner makes mistakes by choosing a bad joint in a low visibility situation, C1 is assigned to the situation and the remedial possible actions that CELTS can take are assigned to C3 (i.e., "Engage dialogue about joint selection").

Given that the situation described by the rule is similar, CELTS can then use the rule to take appropriate action. In short, the previous four questions can be combined to achieve more complex reasoning.

For instance, CELTS detects why some learners do not know which joint must be chosen to achieve a goal (the event "don't_know_right_joint" and "goal#21") (e.g., goal#21= in Fig.14.2.A, moving Canadarm2 from point *A* to the point *End*).

To do so, while CELTS interacts with learners, it seeks all the rules whose left part contains the next information: (don't_know_right_joint, exercise_goal#21). According to the information extracted by EPL, CELTS knows that if the user did not perform distance evaluation and camera adjustment, he/she do not know what is the right joint to be chosen (e.g., goal#21).

The cause may be that the learner forgot to make a distance evaluation or forgot to adjust the camera. According to constraints C3, CELTS can then search the following information: "What is the best camera for observing ISS" or "What is the distance between Canadarm2 and ISS" given the cause found and {don't_know_right_joint,goal#21}. Asking these questions, CELTS helps learners to solve the problem by providing explanations to the learner. This is a form of abductive reasoning. This helps CELTS to predict the results of its action and the learner's response and helps it to choose the best action to help the learner.

14.3.3.2 How CELTS Uses Found Rules to Make Decision

Given that learners make mistakes, after several interactions with learners, CELTS must choose a rule. CELTS consider the left part of the rules as the cause and the right as the effect. However, sometimes several rules are available to answer the same question. For instance, there could be several rules indicating different causes for a mistake made by the learner. To choose a rule, CELTS first computes the *strength* of each rule. We obtained the strength of a rule by multiplying its confidence and support (Faghihi et al. 2010).

The rule with the highest strength will be chosen to help the learner. If two rules have the same strength, then the rules with the highest number of events in the left part matching with the current sequences of events will be selected.

Once CELTS has found the cause, it can ask the learner to confirm or reject the found cause. If the learner deems CELTS answer not to be useful, CELTS will search for the next most likely cause according to the criteria described above.

When the learner rejects an answer, then the support and confidence of the rule decrease. CELTS episodic and causal memories are thus influenced- *the alteration of the support and confidence*. This process will continue until a cause is found. If CELTS cannot find any answer, the message "I am not capable of finding the cause of the problem" is shown.

14.4 Experimental Evaluation of CELTS

We evaluated CELTS from two angles. First, we performed an empirical evaluation with learners to evaluate the following criteria:

1. The number of correct tutoring interventions,
2. The impact of these interventions on learners' performance,
3. The learners' satisfaction,
4. The correctness of the causal rules learned by CELTS.

Second, we analyzed the performance of the data mining algorithms in CELTS and their scalability on larger random databases.

To determine the extent to which the three aforementioned learning mechanisms improved CELTS performance, we asked eight users to test CELTS with learning mechanisms (version *A*) and its use without learning mechanisms (version *B*). Learners were invited to manipulate Canadarm2 for approximately 1 hour, using both versions *A* and *B* of the system. The first four students (group *A*) used version *A*, and then version *B*. The second four learners (group *B*), first used version *B* and then version *A*. After its interactions with the users, CELTS categorized them into novice, intermediate, and expert.

During the trials with version *B*, CELTS automatically learned more than 2400 rules from its interactions with learners. A few examples of rules are found below:

1. Forty two percent of the time, when the learner was not aware of distances and Canadarm2 was closed to the ISS, there was a collision risk,
2. 51% of the time, when a user forgot to adjust the camera, he/she later chose an incorrect joint of Canadarm2,
3. 10% of the time, when the user moved Canadarm2 without adjusting the camera, he/she increased the risk of collisions,
4. 10% of the users who manipulated Canadarm2 close to the space station, being aware of the distance and having reached the goal, were classified as experts,
5. 14% of the time, when the user was inactive, the learner lacked motivation,
6. 25% of the time, when the learner answered three successive questions incorrectly, the learner abandoned the training session,
7. 40% of the time, if the learner adjusted camera 1 and camera 3, but not camera 2, the learner moved the arm close to the space station.

Such rules are then used by CELTS as described in the BN (Fig. 14.1.B) to adapt its behavior to learners. To assure the quality of the rules found by CELTS, we asked a domain expert to evaluate them. Checking all rules one by one would be tedious, the expert examined 150 rules from the 2400+ recorded rules. Overall, the expert confirmed the correctness of about 85 % of the rules. Furthermore, from the found rules, many unexpected rules (e.g., correct) were discovered.

To evaluate to which extent the integration of the mechanisms impacted the performance of the learners; we measured four performance indicators during the usage of version *A* and version *B* of CanadarmTutor by group *A* and group *B*:

1. The percentage of questions they answered correctly,
2. The average time they took to complete each exercise in minutes,
3. The mean number of collision risks incurred by learners,
4. The mean number of violations of the security protocols committed during the exercises.

Fig.14.4 illustrates the next results: 1) Group *A* correctly answered 50% of the questions, whereas Group *B* correctly answered 30% of the questions; 2) Group *A* took an average of 1.45 minutes to complete an exercise, whereas Group *B* took an average of 2.13 minutes; 3) group *A* incurred an average of three collision risks made by learners, whereas Group *B* incurred an average of 5 collision risks made by learners; 4) Group *A* had an average of 10 violated protocols whereas Group *B* had an average of 18 violated protocols.

Although we have not used a very large number of learners in this trial, from these results we see the performance of the learners who used the new version of CELTS clearly improved. Furthermore, we analyzed the correctness of CELTS' hints and messages to the learners during tasks. To determine if an intervention was correct, we asked learners to rate each of CELTS interventions as being appropriate or inappropriate. Because learners could incorrectly rate the tutor's interventions, we also observed the training sessions and verified the ratings given by each learner. The results show the number of appropriate interventions averaged 83 % using version *A* and 58 % using version *B*. This is also a considerable improvement in CELTS's performance over its previous version.

We assessed the user satisfaction by performing a 10 minute post-experiment interview with each user. We asked each participant to tell us which version of CELTS they preferred, to explain why, and to tell us what should be improved.

Users unanimously preferred the new version. Some comments given by users were the following: 1) the tutoring agent "exhibited a more intelligent and natural behavior"; 2) the "interactions were more varied"; 3) the "tutoring agent seems more flexible"; 4) "in general, it gives more appropriate feed-back."

There were also several comments on how CELTS could be improved. In particular, many users expressed the need to give CELTS a larger knowledge base for generating dialogues. We plan to address this issue in future work.

Lastly, to test the scalability of the algorithms, we have generated 20,000 sequences of events automatically. The sequences were generated by randomly answering the questions asked by CELTS during Canadarm2 manipulation.

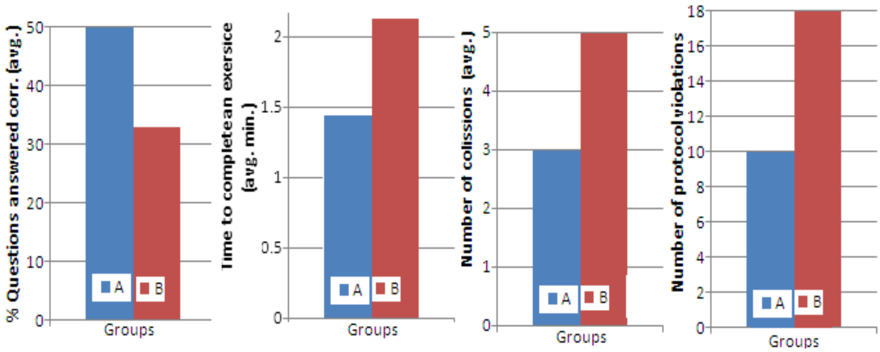


Fig. 14.4 Learners' performance comparison

The data mining algorithms were applied in real time to the sequences to build the causal memory and episodic memory. The performance has been very good, the algorithms terminating in less than 100 seconds with the following parameters: a minimum support of 0.05 and a minimum confidence of 0.3. This demonstrates the scalability of the algorithms for much larger amounts of data than what has been recorded in CELTS during our experiments with users. The aforementioned performance is comparable to the performance of other large-scale frequent pattern mining algorithms which can often handle hundreds of thousands of sequences or transactions (Fournier-viger et al. 2011, Han and Kamber 2006).

14.5 Conclusion

To provide a rich learning experience, an intelligent tutoring agent should be able to take into account past and present events, and should be able to learn from its interactions with learners to continuously improve the assistance it provides. To address this issue, we have presented CELTS, a cognitive tutoring agent which is deployed in a simulator-based tutoring system for learning the complex task of operating the Canadarm2 robotic arm on the ISS.

CELTS is equipped with different types of learning such as emotional, episodic and causal learning. These learning mechanisms have several benefits: 1) the agent remember when a specific learner made a specific mistake; 2) the agent identify the causes of the problem made by the learners; 3) the agent remember positive or negative sequences of event (interactions) from its past experiences.

We have evaluated the new version of CELTS in five ways. First, CELTS performance was evaluated according to the number of correct interventions during training sessions. Second, the results show the new version has a considerable impact on the learners' performance. Third, we have evaluated the satisfaction of users. Fourth, a domain expert has examined the correctness of the causal rules learned by CELTS. Fifth, experiments have confirmed the performance and scalability of the data mining algorithms used in CELTS. For future works, we will further improve CELTS algorithms, the pedagogical strategies used by CELTS and its dialogue generation module.

Acknowledgments. We would like to thank the FQRNT for its financial support and all the members of GDAC who have contributed to the development of CanadarmTutor.

References

Agrawal, R., Imielminski, T., Swami, A.: Mining association rules between sets of items in large databases. In: Proceedings of SIGMOD, pp. 207–216. ACM Press, Washington, D.C. (1993)

Anderson, J.R.: Rules of the mind. Erlbaum, Hillsdale (1993)

Anderson, J.R., Bothell, D., Byrne, M.D., Douglass, S., Lebiere, C., Qin, Y.: An integrated theory of the mind. Psychological Review 111(4), 1036–1060 (2004)

Baars, B.J.: In the theater of consciousness: The workspace of the mind. Oxford University Press, New York (1997)

Braun, M., Rosenstiel, W., Schubert, K.D.: Comparison of Bayesian networks and data mining for coverage directed verification category simulation-based verification. In: Proceedings of 8th High-Level Design Validation and Test Workshop of IEEE (2003)

Brusilovsky, P., Millán, E.: User Models for Adaptive Hypermedia and Adaptive Educational Systems. In: Brusilovsky, P., Kobsa, A., Nejdl, W. (eds.) Adaptive Web 2007. LNCS, vol. 4321, pp. 3–53. Springer, Heidelberg (2007)

Cannon, W.: The James-Lange theory of emotion: A critical examination and an alternative theory. American Journal of Psychology 39(2), 10–124 (1927)

Damasio, A.: The feeling of what happens: Body and emotion in the making of consciousness. Harvest Books, New York (2000)

Deogun, J.S., Jiang, L.: Prediction Mining – An Approach to Mining Association Rules for Prediction. In: Ślęzak, D., Yao, J., Peters, J.F., Ziarko, W.P., Hu, X. (eds.) RSFDGrC 2005, Part II. LNCS (LNAI), vol. 3642, pp. 98–108. Springer, Heidelberg (2005)

Dubois, D., Poirier, P., Nkambou, R.: What does consciousness bring to CTS. In: Proceedings of AAAI. LNCS, vol. 5091, pp. 803–806. Springer, Berlin (2007)

Faghihi, U., Fournier-Viger, P., Nkambou, R., Poirier, P.: A Generic Episodic Learning Model Implemented in a Cognitive Agent by Means of Temporal Pattern Mining. In: Chien, B.-C., Hong, T.-P., Chen, S.-M., Ali, M. (eds.) IEA/AIE 2009. LNCS (LNAI), vol. 5579, pp. 545–555. Springer, Heidelberg (2009)

Faghihi, U., Fouriner-viger, P., Nkambou, R., Poirier, P.: The Combination of a Causal and Emotional Learning Mechanism for an Improved Cognitive Tutoring Agent. In: García-Pedrajas, N., Herrera, F., Fyfe, C., Benítez, J.M., Ali, M. (eds.) IEA/AIE 2010, Part II. LNCS, vol. 6097, pp. 438–449. Springer, Heidelberg (2010)

Faghihi, U., Poirier, P., Fournier-Viger, P., Nkambou, R.: Human-like learning in a conscious agent. Journal of Experimental & Theoretical Artificial Intelligence 23(4), 497–528 (2011)

Fournier-Viger, P., Nkambou, R., Tseng, V.S.: RuleGrowth: mining sequential rules common to several sequences by pattern-growth. In: Chu, W.C., Wong, W.E., Palakal, M.J., Hung, C. (eds.) Proceedings of SAC, pp. 954–959 (2011)

Franklin, S., Patterson, F.G.J.: The LIDA architecture: Adding new modes of learning to an intelligent, autonomous, software agent. In: Proceedings of IDPT (2006)

Gopnik, A., Glymour, C., Sobel, D.M., Schulz, L.E., Kushnir, T., Danks, D.: A theory of causal learning in children: Causal maps and bayes nets. Psychological Review 111(1), 3–32 (2004)

Han, J., Kamber, M.: Data mining: Concepts and techniques. Morgan Kaufmann, San Francisco (2006)

Hélie, S.: Modélisation de l'apprentissage ascendant des connaissances explicites dans une architecture cognitive hybride. Ph.D Dissertation, Université du Québec à Montréal (2007)

Hipp, J., Güntzer, U., Nakhaeizadeh, G.: Data mining of association rules and the process of knowledge discovery in databases. In: Proceedings of ICDM, pp. 15–36. Springer, London (2002)

James, W.: What is an emotion. Mind 9, 188–205 (1884)

LeDoux, J.E.: Emotion circuits in the brain. Annual Review of Neuroscience 23, 155–184 (2000)

Li, D., Deogun, J.S.: Discovering Partial Periodic Sequential Association Rules with Time Lag in Multiple Sequences for Prediction. In: Hacid, M.-S., Murray, N.V., Raś, Z.W., Tsumoto, S. (eds.) ISMIS 2005. LNCS (LNAI), vol. 3488, pp. 332–341. Springer, Heidelberg (2005)

Maes, P.: How to do the right thing. Connection Science 1, 291–323 (1989)

Nason, S., Laird, J.E.: Soar-RL: Integrating reinforcement learning with Soar. Cognitive Systems Research 6(1), 51–59 (2005)

Nkambou, R., Belghith, K., Kabanza, F.: An Approach to Intelligent Training on a Robotic Simulator Using an Innovative Path-Planner. In: Ikeda, M., Ashley, K.D., Chan, T.-W. (eds.) ITS 2006. LNCS, vol. 4053, pp. 645–654. Springer, Heidelberg (2006)

Phelps, E.A.: Emotion and cognition: Insights from studies of the human amygdala. Annual Review of Psychology 57, 27–53 (2006)

Purves, D., Brannon, E., Cabeza, R., Huettel, S.A., LaBar, K., Platt, M., Woldorff, M.: Principles of cognitive neuroscience. Sinauer Associates, Sunderland (2008)

Rolls, E.T.: Neurophysiology and functions of the primate amygdala, and the neural basis of emotion. In: Aggleton, J.P. (ed.) The Amygdala: A Functional Analysis, pp. 447–478. Oxford University Press, London (2000)

Schoenbaum, G., Chiba, A.A., Gallagher, M.: Changes in functional connectivity in orbitofrontal cortex and basolateral amygdala during learning and reversal training. The Journal of Neuroscience 20(13), 5179–5189 (2000)

Schoppek, W.: Stochastic independence between recognition and completion of spatial patterns as a function of causal interpretation. In: Proceedings of ACCSS (2002)

Sherry, D.F., Schacter, D.L.: The evolution of multiple memory systems. Psychological Review 94, 439–454 (1987)

Squire, L.R., Kandel, E.R.: Memory: From mind to molecules. Scientific American Library, New York (2000)

Sun, R.: Duality of the mind: A bottom-up approach toward cognition. Erlbaum, Hillsdale (2001)

Sun, R.: The CLARION cognitive architecture: Extending cognitive modeling to social simulation. In: Sun, R. (ed.) Cognition and Multi-Agent Interaction, pp. 79–99. Cambridge University Press, New York (2006)

VanLehn, K.: Student modeling. In: Polson, M., Richardson, J. (eds.) Foundations of Intelligent Tutoring Systems, pp. 55–78. Erlbaum, Hillsdale (1988)

VanLehn, K., Lynch, C., Schulze, K.G., Shapiro, J.A., Shelby, R., Taylor, L., Treacy, D., Weinstein, A., Wintersgill, M.: The Andes physics tutoring system: Lessons learned. Journal of Artificial Intelligence in Education 15(3), 147–204 (2005)

Vernon, D., Metta, G., Sandini, G.: A survey of artificial cognitive systems: Implications for the autonomous development of mental capabilities in computational agents. IEEE Transactions on Evolutionary Computation 11(2), 151–180 (2007)

Westen, D.: Psychology: Mind, brain, culture. John Wiley & Sons, New York (1999)

Woolf, B.P.: Building intelligent interactive tutors: Student centered strategies for revolutionizing e-learning. Morgan Kaufmann, Boston (2009)

Abbreviations

ACT-R	Adaptive Control of Thought-Rational
AIED	Artificial Intelligence in Education community
BN	Behavior Network
CELTS	Conscious-Emotional-Learning Tutoring System
CLARION	Connectionist Learning with Adaptive Rule Induction ON-line
CLM	Causal Mechanism
CR	Collision risk
EM	Emotional Mechanism
EML	Emotional Learning
EPL	Episodic Learning
EPM	Episodic Mechanism
ISS	International Space Station
ITS	Intelligent Tutoring Systems
WM	Working Memory

Part IV
Applications

Chapter 15
Incorporation of Agent Prompts as Scaffolding of Reflection in an Intelligent Learning Environment

Longkai Wu and Chee-Kit Looi

National Institute of Education
1 Nanyang Walk, Singapore
{longkai.wu,cheekit.looi}@nie.edu.sg

Abstract. Recent research has emphasized the importance of reflection for students in an intelligent learning environment. But, researchers have not reached a consensus on the most effective ways to design scaffolding to prompt reflection, nor have they accepted a common mechanism that can explain the effects of scaffolding on reflection. Two types of agent prompts to foster reflection are contrasted in this chapter, both from the perspective of a tutee, differing in their specificity. Generic prompts are content-independent tutee questions, aiming at fostering students' reflection on metacognitive strategies and beliefs regarding their learning-by-teaching activities. Specific prompts, on the other hand, are content-dependent tutee questions that encourage students' reflection on domain-related and task-specific skills, and articulation of their explanatory responses. This chapter describes the design and effect of these two types of agent prompts, adapted to students' learning-by-teaching activities, on the learning outcomes, the elicited levels of reflection, and the self-efficacy of the secondary school students.

15.1 Introduction

The educational research literature suggests that questions prompts, whether from teachers, peers or textbooks, could promote reflection by eliciting explanations. (Rothkopf 1966) investigates the ways in which questions inserted in texts affected subjects' understanding of the texts. (Chi et al. 1994) indicate that questions that elicit self-explanations lead to improved understanding of texts. And students, who provide explanations to other students' questions or explain examples found in their textbooks, seem to strengthen connections among their ideas (Davis 1998). (Moon 2004) further suggests structuring reflection with questions to deepen the quality of reflection.

A. Peña-Ayala (Ed.): Intelligent and Adaptive ELS, SIST 17, pp. 369–391.
springerlink.com

Researchers in Intelligent Learning Environment (ILE) have recognized the importance of incorporating question prompts into ILE design (Hmelo and Day 1999). Question prompts are used as scaffolds to help direct students towards learning-appropriate goals, such as focusing student attention and modeling the kinds of questions students should be learning to ask (Azevedo and Hadwin 2005). Positive evidences are found for question prompts to help students with various aspects, such as, knowledge integration (Davis and Linn 2000) and ill-structured problem-solving processes (Ge and Land 2004; Xie and Bradshaw 2008). Meanwhile, mechanisms for supporting self-explanation, tutorial dialog or reflective dialog (Aleven and Koedinger 2002; Grassser et al. 2001; Katz et al. 2000) have been prevalent in traditional intelligent tutoring systems (ITS), in which the computer plays the role of tutor (e.g., Cognitive Tutor, AutoTutor).

This chapter is concerned with the investigation of agent prompts (i.e., question prompts to initiate learners' reflection in learning, within an agent-enabled learning-by-teaching context). The mechanism of generating agent prompts defines how to assess and model learners' reflective learning-by-teaching activities and metacognitive skills. We intend to explore how a learning partner, acting as the role of inquisitive tutee enabled by the generation of question prompts, might be used to address the challenge of facilitating reflection in a student tutor linked to learning-by-teaching activities. Here, the reflection of the student tutor mainly refers to an intermingled process of knowledge construction and metacognition as a direct result of his engagement in instructional activities inherent to the virtual tutoring process, such as explaining, answering questions from the tutee, correcting errors of the tutee and asking questions to the tutee (Cohen 1986; Garneret, et al. 1971; King 1998). The opportunity for reflection enables the tutor to monitor his own understanding, recognize and repair knowledge gaps and misconceptions, integrate new knowledge with prior knowledge, and generate new ideas for self-evaluation and reflection (Roscoe and Chi 2007).

15.2 Review of Literature

The section firstly reviews how question prompts are used as the scaffolding approach to enhance students' reflective learning. Then we introduce tutee questions as types of question prompts to facilitate reflection and learning. An overview of metacognitive and cognitive strategies and beliefs is subsequently included, which leads the differentiation of generic prompts and specific prompts, in the context of tutee questions, that we are intended to investigate in this study.

15.2.1 Question Prompts

Classroom studies have suggested that prompts fostering reflection could be effective because they provide support for the cognitively complex ways learners think about, feel, and make connections in experience (Davis and Linn 2000). By engaging in reflective activities such as responding to the reflection questions, the learner builds their understanding and locates the significance of his activity in a larger context. Thus he is enabled to observe the meaning, he has taken from the experience and excavate the underlying qualities that make the experience significant. When the learner is prompted to deeper forms of reflection, it also becomes possible for him to identify learning edges, those questions or issues that he is seeking to understand in order to advance his work (Amulya 2004). In doing reflection stimulated by prompts, the learner can unpack the richness of the experience and evaluate which issues emerging from that experience need to be pursued.

15.2.2 Tutee Questions

Meanwhile, recent research also shows the evidence of learning benefits to tutors from tutee's question prompts in the context of peer tutoring. (Cohen et al. 1982) demonstrate empirical evidence of learning gains for tutors compared to non-tutors in the context of peer tutoring. (King et al. 1998) specially study the tutor's explanations and questioning in the tutoring process as the sources for tutor's learning based on high-level question stems (i.e., questions prompting for comparisons, justifications, causes-and-effects, evaluations, etc.). (Graesser et al. 1995) show that the tutee's occasional "deep" questions out of major "shallow" questions can stimulate the tutor's deeper response. (Coleman et al. 1997) demonstrate very similar findings in collaborative learning settings with students using high-level explanation prompts. (Roscoe and Chi 2004) find that in a non-reciprocal and naturalistic (i.e., little or no training) tutoring context, the tutee's questions can motivate tutor explanations and metacognition, and thus have a significant and positive influence on the tutor's learning activities and opportunities.

(Graesser et al. 1995) discuss the kinds of tutee questions that occur during tutoring, which can be divided into shallow and deeper questions. Shallow factual questions ("what" questions) ask definitions or simple calculations while deeper questions ("how" and "why" questions) ask about causal relationships and underlying principles, requiring elaboration, inference and logical reasoning. Peverly and Wood 2001 indicate that deeper questions support learning more efficiently than shallow questions.

15.2.3 Cognitive and Metacognitive Strategies

(Cornford 2002) notes that cognitive strategies and metacognitive strategies are closely related since both of them involve cognition and skill but conceptually they are quite distinct. Cognitive strategies are used to help an individual achieve a particular goal (e.g., understanding a text) while metacognitive strategies are used to ensure that the goal has been reached (e.g., quizzing oneself to evaluate one's understanding of that text) (Cornford 2002). (Weinstein and Meyer 1991) state: "A cognitive learning strategy is a plan for orchestrating cognitive resources, such as attention and long-term memory to help reach a learning goal". They indicate that there are several characteristics of cognitive learning strategies, including that they are goal-directed, intentionally invoked, and effortful that are not universally applicable, but situation specific.

Comparatively, (Schraw 1998) notes that metacognitive strategies appear to share most of these characteristics, with the exception of the last one, since they involve more universal application through focus upon planning for implementation, monitoring and evaluation. It means that metacognitive strategies are not so situation specific but, involve truly generic skills essential for learner, more sophisticated forms of thinking and problem solving.

15.2.4 Generic and Specific Prompts

The generic prompts, or called "general tutee questions" are a series of content-independent questions to lead students to reflect on metacognitive strategies and beliefs in learning and teaching, consider various perspectives regarding their activities, such as "Why should you teach?", "Before starting to teach, can you think about what you are supposed to learn from it?", "What do you learn from me as your tutee?". (VanLehn, Jones and Chi 1992) suggest: "Gneric prompts could increase the chances that individual students will be able to identify gaps in their own understanding, discover deficiencies in their mental models, or generate useful inferences."

The specific prompts, or called "specific tutee questions", on the other hand, are a series of content-dependent questions to lead students to reflect on task-specific and domain-related skills regarding their activities and to articulate their explanatory responses, such as "Can you explain the concepts you just taught me?", "If you query me by asking me a casual question in the below window, you can see how I reason through the concept map that you have taught me. Can you tell me if my reasoning process is correct and give me a further explanation?" Specific prompts appear to be helpful in getting some students to realize that they have a gap in their understanding and may even hint at how to fill the gap (Aleven et al. 2006; VanLehn et al. 1992).

15.3 System Design

This section describes the design and implementation of agent prompts within the Betty's Brain system with both pedagogical and technological considerations. This research has been concerned with the creation of agent prompts that explores new scaffolding approaches in an intelligent learning-by-teaching environment. We puts forth the scheme and architecture of an agent prompts generator which produces two types of prompts to guide students' reflection within a learning-by-teaching environment. The purpose of designing agent prompts generator is to enhance the learning-by-teaching environment, reifying the metacognitive, task-specific and domain-related reflection involved in such activities.

15.3.1 Overview

The development work focused on the generation and incorporation of meaningful agent prompts, which can arouse student's reflective learning-by-teaching activities. The work was built on an existing system, Betty's Brain (Fig. 15.1), a learning-by-teaching agent environment built by the Teachable Agent Group in Vanderbilt University (Biswas et al. 2001). With the ability to learn what the students have taught by concept mapping, Betty's Brain was used to play the role of agent tutee in our research.

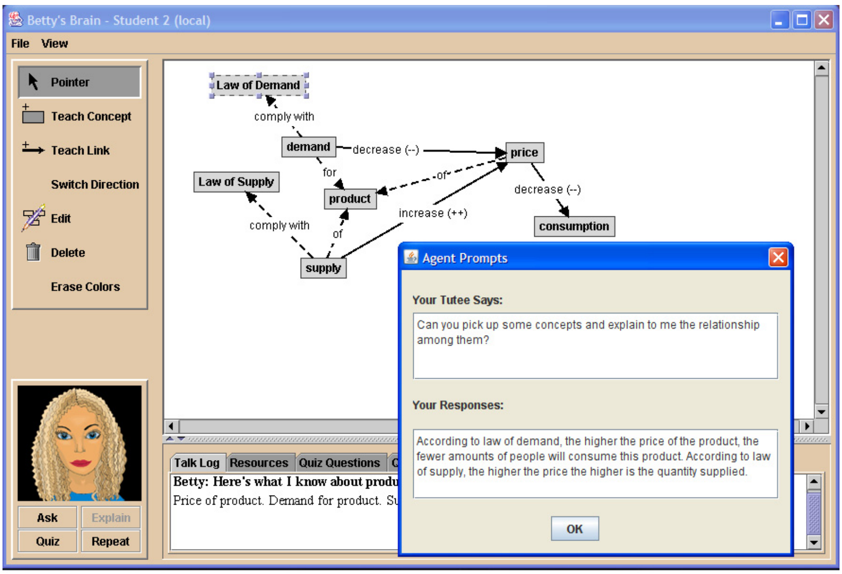


Fig. 15.1 Adapted Betty's Brain (Biswas, et al. 2011) in basic economics

We adapted Betty's Brain to become an inquisitive agent tutee, with the built-in agent prompts generator, in the domain of basic economics. When students interact with the inquisitive agent tutee version of Betty's Brain, they have to respond to the question prompts from the agent system. These question prompts were produced in the agent prompts generator, which could analyze the semantic structure of student concept maps and compare them with expert maps tailored in the domain of basic economics. Our goal was to foster a reflective student-agent learning-by-teaching interaction for better learning outcomes.

15.3.2 Aspects of Consideration

The consideration of generating appropriate question prompts is to guide students in learning-by-teaching activities, by provoking different reflection types in their responses and exploring situational cues and underlying meanings relevant to the context. Students were expected to recognize the importance of reflective activities while, their cognitive load was not significantly increased.

Based on the literature review and the design principles of an agent tutee system, three aspects (Fig. 15.2) were considered, when the agent prompts scheme was designed: Learning-by-Teaching Stages, Reflection Types and Patterns in Student Maps.

Aspects	Value
Learning-by-Teaching Stages	Familiarization Production Evaluation Post-Task Reflection
Reflection Types	Double-Loop Reflection Single-Loop Reflection
Patterns in Student Maps	Missing or Incorrect Expert Concepts Missing or Incorrect Expert Propositions

Fig. 15.2 Aspects of consideration

15.3.3 Learning-by-Teaching Stages

Our learning-by-teaching activities were categorized into four stages as shown in Fig. 15.3, which follow the conceptual stages in practicing tutoring with a metacognition instruction model that focuses on the following metacognitive skills: 1) problem understanding and knowledge monitoring; 2) selection of metacognitive strategies; 3) evaluation of the learning experience (Gama 2004).

- Stage 1 Familiarization, understanding, and planning: This stage contains two types of reflective activities that include self-assessment of the understanding of the domain knowledge and difficulties and self-selection of metacognitive strategies,
- Stage 2 Production, teaching, presenting answer, and answers: This stage is devoted to enable students to teach the agent tutee what they have learned to the agent tutee by constructing concept maps and monitoring the agent tutee's understandings,
- Stage 3 Evaluation, evaluating the performance: This stage provides students with opportunities to evaluate the performance of the agent tutee, as well as their own performance,
- Stage 4 Post-Task Reflection, Reflecting on Learning-by-Teaching Experience: This stage is oriented to promote post-practice reflection on the learning-by-teaching processes and the strategies implemented.

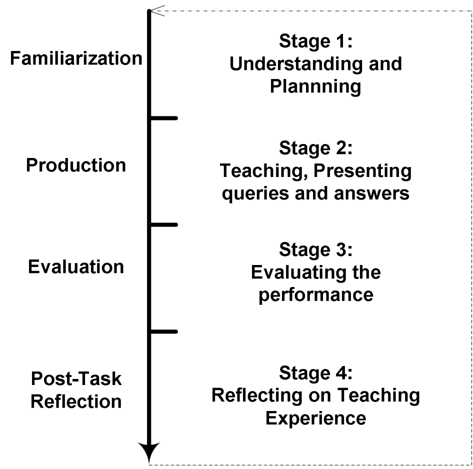


Fig. 15.3 Stages of learning-by-teaching activities

15.3.4 Reflection Types

Considering both the theoretical and practical perspectives, we design agent prompts to foster two major types of reflection for students in the learning-by-teaching environment.

- Generic prompts fostering double-loop reflection are content-independent, stimulating students to monitor their learning-by-teaching processes and consider various perspectives regarding their learning-by-teaching activities. Double-loop reflection focuses on the examination and reflection of the theory or perspective in use,
- Specific prompts fostering single-loop reflection are content-dependent, providing students with a structure through the learning-by-teaching process and lead them to complete a specific cognitive task and articulate their explanatory responses. Single-loop reflection refers to increasing efficiency of an objective, which is task oriented and is about the design of the process to retain reliability.

The content of sample generic prompts containing metacognitive strategies and beliefs, which was developed from the adaptation of Reflection Assistant Model (Gama 2004) attempting to simulate a tutee's perspective, is depicted as follows.

- List of strategies for assessing beliefs:
  - What do you think about teaching and who is it for?
  - Why should you teach?
  - What do you think about what you are supposed to learn from teaching?
  - What do you learn from me as your tutee?

- List of strategies for monitoring understanding:
  - Can you read your learning objectives more than once?
  - Can you read the learning objectives to separate the important parts?
  - Can you think of a related learning task you have already done and use it as an example?
  - Before starting to teach, can you think about what you are supposed to learn from it?
  - Can you read the learning objectives and determine which parts you don't understand well?
  - Can you review the basic concepts that are not clear, before you begin to teach?
  - Can you set a goal, and think about the steps to reach this goal?
- List of strategies for controlling errors:
  - Can you stop and review each part in the map to see if a mistake has been made?
  - Can you reread the resources to check for missing important parts?
  - Can you change strategies if you feel lost and confused and don't seem to move anywhere?
- List of strategies for revising:
  - Can you think about a way of checking to see if your map is correct?
  - Can you review all things done to make sure nothing is missing?
  - Can you reread the learning objectives and resources and ask if the map really meets the description in the learning objectives and resources?

The content of sample specific prompts containing domain-related and task-specific skills, which was developed from the adaptation of Teachable Agent (Leelawong 2005), was partly illustrated as follows.

- Read the on-line resources to learn:
  - Can you check the on-line resources for more information and tell me more?
- Request explanation on concepts or propositions:
  - Can you explain the concepts you just taught me?
  - Can you pick up two concepts from that section and explain to me the relationship among them?
- Query to teach better:
  - A good teacher asks students questions to make sure that they understand things correctly. You can ask me by clicking on the *Ask* button underneath the window. Let me know if my answer is useful by offering a description.

- Query causal questions in order to teach:
  - If you ask me a casual question, you can see how I reason through the concept map that you have taught me. Can you tell me if my reasoning process is correct and give me a further explanation?
- Ask for quiz in order to teach:
  - I have learned something from you. Please require me to take a quiz. Can you give an evaluation comment on my quiz performance?

15.3.5 Patterns in Student Maps

Our intention is to develop an evaluation scheme for student concept maps, which supports and facilitates the identification and categorization of faulty propositions in a concept map.

We classify learners' errors based on the pattern categories that are presented below, including missing or incorrect expert concepts and expert propositions. Moreover, this categorization is used as the basis for the construction of agent prompts that reflect the different types of student errors.

- Missing expert concepts:
  - The student omits specific concepts (which are considered fundamental concepts of the subject matter) from their maps. The usual omissions of specific concepts lead us to the conclusion that the student manifests incomplete understanding.
- Incorrect Expert Concepts:
  - The students make mistakes on specific concepts (which are considered fundamental concepts of the subject matter) in their concept maps. The usual mistakes of specific concepts lead us to the conclusion that the student manifests incorrect understanding.
- Missing Expert Propositions: The student uses specific relationships between two or more concepts, which are not false but they do not correctly/fully address the relation of these concepts in the context of the subject matter. He does not relate two or more concepts denoting their relationship. These cases are considered as an evidence of incomplete understanding.
- Incorrect Expert Propositions: The student relates two or more concepts that should not be related, and/or with incorrect relationships that lead to clearly false propositions. The mistakes of expert propositions lead us to the conclusion that the student manifests incorrect understanding.

15.3.6 Architecture of Agent Prompts Generator

In Fig. 15.4, the architecture of Agent Prompts Generation System is depicted with three major components involved, namely the Agent Prompts Generator, the Map Comparer, and the Stage Detector.

15.3.6.1 Agent Prompts Generator

The Agent Prompts Generator monitors the student's concept mapping activities (i.e., the student teaches Betty to tailor a concept map in the agent environment) and plays the role of coordinator in the system. It mainly receives the results from the Map Comparer, selects proper prompts from the repository of question prompts and sends them to the Reflective Dialogue window for students to respond to. The student tutor receives these prompts in the Reflective Dialogue window and responds while, teaching the agent by modeling in a Concept Map Editor.

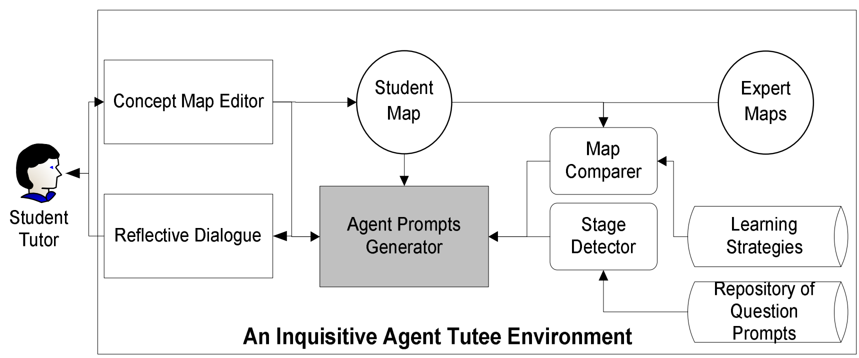


Fig. 15.4 Architecture of Agent Prompts Generator

15.3.6.2 Map Comparer

The map comparer detects patterns and assesses student performance by comparing student map to an integrated expert map. It selects proper prompting strategies for an individual student from the repository of learning strategies, which stores tactics and strategies specified to the agent environment (e.g., examining agent's understanding, observing the agent's independent performance, etc.), based on his concept mapping practice.

An overlay assessment method is used to evaluate students by matching their student maps with expert maps. To be accurate for measurement, an integrated expert map is considered by integrating a number of expert maps developed by separate experts (teachers and researchers). We adopt a fuzzy integration technique (Chen et al. 2001) based on fuzzy set discipline that attempts to produce an integrated expert map that could be superior to any of the individual expert map.

The algorithm for detecting patterns in student concept maps, with a breadth-first search through the expert map (Kornilakis et al. 2004), begins at the central concept ("Demand" concept in our case). A queue is used to collect the concepts that have not yet been searched. The algorithm appears as in the next pseudo code.

Combining this with the breadth-first nature of the algorithm, we can be certain that it will always be possible to find the student concept node corresponding to the expert concept node in the expert map.

The algorithm is also guaranteed to end either after finding a pattern or after confirming that the student map matches the expert map.

The correspondence between map patterns and agent prompts was pre-defined based on the design principles within an Adaptive Feedback Framework (Gouli et al. 2005). Table 15.1 shows the samples of patterns found in student maps and appearing as corresponding agent prompts stored in the repository of question prompts.

```
Insert central concept in queue
While the queue is not empty repeat
v <-the first concept in the queue
v' <-the concept corresponding to v in the expert map
E <-the set of all links coming out of student map
E' <-the set of all links coming out of expert map
For every link e' in E'
  d' <-the destination concept of e'
  If e' not in E
     If there exists a link in E ending on d'
       If there exists a link in the expert map from n'
                to d' other than e'
          Return Missing Expert link
       Else
         Return Incorrect Expert Link
     Else
       Search the student map for d'
       If found
          Return Missing Expert Link
       Else
          Return Missing Expert Concept
  Else
    e <-the link corresponding to e' in the student map
    If the destination concept of e is not d'
       If there exists a concept other that d'
           in the expert map connected to n' with a link e'
          Return Incorrect Expert Concept
       Else
          Return Missing Expert Concept
          For every link e in E
            If e not in E'
               Return Missing Expert Link
```

Table 15.1 Agent Prompts Generation based on Pattern Detection

Pattern	Description
Missing expert concepts	Missing concept and its relationships when specific concepts defined in concept map are omitted from the student map E.g. The concept of "Income" and its links with the concepts of "Demand" are missing from the student map. **Demand** Generic prompt: "*Can you review all things to check for missing important parts to teach me and give me an explanation*?" Specific prompt: "*Do you consider that you could teach me the concept of 'Income' and give me an explanation*?"
Incorrect expert concepts	A concept is related to an incorrect concept which should be replaced with another concept. E.g. In the proposition "Demand *affects* Elasticity", the concept "Elasticity" should replace the concept "Electricity". **Demand** $\xrightarrow{\textbf{Affect}}$ **Electricity** Generic prompt: "*Can you stop and review each part in the map to see if you have made a mistake and give me an explanation*?" Specific prompt: "*Do you want to reconsider the concept of 'Electricity' you have taught me and give me an explanation*?"
Incorrect expert propositions	Two concepts are related even though they should not be. E.g. The proposition "Opportunity Cost *determines* Demand" is incorrect, as the concepts "Opportunity Cost" and the "Demand" are not related. So, the link "determine" should be omitted. **Opportunity Cost** $\xrightarrow{\textbf{Determine}}$ **Demand** Generic prompt: "*Can you stop and review each part in the map to see if a mistake has been made and give me an explanation*?" Specific prompt: "*Do you want to reconsider the link of 'Determine' between 'Opportunity Cost' and 'Demand' and give me an explanation*?"
Missing expert propositions	The proposition of two expert concepts is missing on the student concept map. E.g. The concepts "Income" and the "Demand" are not related although they should be linked with the relationship "increase". **Income** **Demand** Generic prompt: "*Can you review all things done to make sure nothing is missing and give me an explanation*?" Specific prompt: "*Do you want to consider teaching the link between 'Income' and 'Demand' again and give me an explanation*?"

15.3.6.3 Student Performance Assessment

We use an overlay assessment method proposed by Chang et al. (2005) to compare student concept maps with expert concept maps in order to assess student performance. By comparing concept maps drawn by the students with the expert concept maps, the students' comprehension of each proposition can be determined.

A student's comprehension has one of the following learning states: the proposition is learned (complete expert concepts and expert propositions), partially learned (missing expert concepts or missing expert propositions), unlearned (no expert concepts or no expert propositions), or the student has a misconception about the proposition (incorrect expert concepts or incorrect expert propositions).

The learning state of the student revealed in his map is used to grade student's performance and determine the frequency of agent prompts delivery. Once the similarity between student map and expert map is high enough, the agent prompts are removed just as traditional scaffolds were faded.

The proposed method considers propositions based on weighted concept maps.

Let $G_e = (V_e, E_e)$ be an expert concept map. If $(v_i, v_j) \in V_e$ and $e_{ij} \in E_e$, then (v_i, e_{ij}, v_j) represents a proposition in G_e if the relation link e_{ij} connects two concept nodes v_i and v_j. Any proposition (v_i, e_{ij}, v_j) can be compared with the propositions in a student concept map. From the resulting comparison, it is possible to decide if the proposition (v_i, e_{ij}, v_j) is learned, partially learned, unlearned, or if the student has a misconception. The following procedure shows how the comparison is performed:

- If there is a proposition (v_i, e^*_{ij}, v_j) in the student concept map, then
  - If $e^*_{ij} = e_{ij}$, then (v_i, e_{ij}, v_j) is learned.
  - If $e^*_{ij} = \phi$, then (v_i, e_{ij}, v_j) is partially learned.
  - If $e^*_{ij} \bullet e_{ij}$, then the student has misconception about (v_i, e_{ij}, v_j).
- If there is a proposition (v_j, e^*_{ij}, v_i) in the student concept map, then
  - If $e^*_{ij} = e_{ij}$ or $e^*_{ij} = \phi$, (v_i, e_{ij}, v_j) is partially learned.
  - If $e^*_{ij} \bullet e_{ij}$, the student has misconception about (v_i, e_{ij}, v_j).
- If the proposition (v_i, e^*_{ij}, v_j) or (v_j, e^*_{ij}, v_i) does not exist in the student concept map, then (v_i, e^*_{ij}, v_j) is not learned.

In order to quantify the similarity between student map and expert map, all student propositions scores according to the student's learning state are calculated and then the similarity index S is computed by the means of equation 15.1.

$$S = \frac{\sum_{i,j} score(v^*_{p_i})}{\sum_{v_{p_i} \in V_p} w(v_{p_i})}, 0 \le S \le 1 \qquad (15.1)$$

In this equation, v_{p_i} is a proposition in the expert map and $w(v_{p_i})$ is its weight. The $score(v^*_{p_i})$ is the score assigned to the proposition $v^*_{p_i}$. After calculating the scores for all student propositions, the similarity index S is achieved and used to measure how similar the student's knowledge structure is to the expert's. The larger the index, greater is the similarity. Once S equals 1, the agent prompts is totally removed to let students fully concentrate on the concept mapping activities.

15.3.6.4 Stage Detector

Apart from Map Comparer, the Agent Prompts Generator also needs the Stage Detector to detect the student's learning-by-teaching stage as discussed in the previous sections, which includes Familiarization, Production, Evaluation and Post-Task Reflection, and stimulate student in different cognitive and metacognitive aspects of reflection in learning.

The Stage Detector detects the stage in which the student is involved and retrieves and selects the appropriate agent prompts within the repository of question prompts. Examples of question prompts stored in the repository adapted to the four stages are the following.

- Agent prompts in the familiarization stage:
  - Generic prompts: What do you think about teaching and who is it for?
  - Specific prompts: How will you comment on the materials you will teach me at the beginning?
- Agent prompts in the production stage:
  - Generic prompts: How do you want me, as your tutee, to deal with you?
  - Specific prompts: It seems you have taught me several concepts. Can you choose some to explain to me?
- Agent prompts in the evaluation stage:
  - Generic/specific prompts: What is the most important thing you have tried to teach me?
- Agent prompts in the post-task reflection stage:
  - Generic prompts: What is your thinking after teaching Betty?
  - Specific prompts: Can you provide your further comments on the evaluation on Betty (advantages, weaknesses, expectations, etc…)?

15.4 Empirical Study

This study is to investigate whether the proposed framework is able to help learners in reflection and learning.

15.4.1 Participants and Procedure

Participants were 33 students from two grade levels (level 1 and 2) in two local secondary schools (ages ranged from 13 to 15), who took part in the experiments on a voluntary basis for two-hour sessions within 2-week period (Table 15.2).

Table 15.2 Procedure of Empirical Study

Phases	Activities	Description
Phase 1	Pre-test	MSLQ and Knowledge Pre-test
Phase 2.1	Tutoring: Familiarization	Get familiar with materials and simulated tutee
Phase 2.2	Tutoring: Production	Teach simulated tutee by concept mapping
Phase 2.3	Tutoring: Evaluation	Check the performance of simulated tutee
Phase 2.4	Tutoring: Post-Task Reflection	Reflect upon own performance
Phase 3	Post-test	MSLQ and Knowledge Post-test (1 week later)

They were randomly assigned to one of the three conditions to study elementary economics of demand and supply. Economics is a theoretical and applied domain, seldom studied in secondary school and rarely adopted as the domain in ILE research. The domain materials were provided to participants before the sessions.

In short, 29 students (76%), 20 female (69%) and 9 male (31%) completed all activities within the 2-week period of the trial, resulting in the next division over the three conditions: no prompts (NP) condition as control group: n = 10, specific prompts (SP) condition: n = 10 and generic prompts (GP) condition: n = 9.

During the tutoring phases, participants were working with the simulated tutee system to teach what they learnt from materials by constructing concept maps. The NP group (n=10) worked with the basic version of simulated tutee without prompts. The SP group (n=10) worked with the version embedded with specific prompts. The GP group (n=9) worked with the version embedded with generic prompts. Both the SP and GP groups were required to write down their reflection statements in the dialog window to respond to the simulated tutee prompts to proceed with their tutoring activities. A sample of response statements from participants to two types of prompts are as follows:

```
…
[Simulated tutee detects decreasing of missing expert propositions in the production phase] ¨
Can you pick up some concepts and explain to me the relationship among them? (Specific Prompts)
[SP Student] According to law of demand, the higher the price of the product, the fewer amounts of people will consume this product. According to law of supply, the higher the price the higher is the quantity supplied
…
[Simulated tutee detects start of the post-reflection phase]
What is your thinking after teaching me? (Generic Prompts)
```

```
[GP Student] You are a curious student by asking a lot of
questions to me. But sometimes, I don't quite understand what
you are asking me to do. I need to learn more about demand
and supply to teach you better.
...
```

We further categorized the participants into High and Low group according to their self-efficacy scores in their MSLQ (Motivated Strategies for Learning Questionnaire) pre- and post- test. Participants scored above the mean self-efficacy score in MSLQ pre-test were included in the High group and the rest were included in the Low group.

Zimmerman (2000) notes that self-efficacy has emerged as a highly effective predictor of students' motivation and learning. As a performance-based measure of perceived capability, self-efficacy differs conceptually and psychometrically from related motivational constructs, such as outcome expectations, self-concept, or locus of control. We selected nine questions from the MSLQ developed by Pintrich and DeGroot (1990), in which Questions 2, 6, 8, 9, 11, 13, 16, 18, and 19 are serving the purpose of determining one's self-efficacy. These questions helped us to avoid subject bias or subject characteristics threat by clarifying students' tendencies in learning activities.

15.4.2 Impact of Question Prompts on Knowledge Pre-/Post-Test

Fig. 15.5 uses error bars of the pre-test and post-test scores to represent the pre-test/post-test scores across groups. It indicates that there was a tendency for learners in all three experimental groups to achieve approximately the same level in the pre-test while both the two prompted conditions (GP and SP) outperformed the non-prompted condition in the post-test significantly (the result of ANOVA test is $F(2, 25) = 19.55, p < .05$).

We also compared the pre-test-to-post-test effect sizes (Cohen's d) of the three conditions. As seen in the Table 15.3, the two prompted conditions yielded an average effect size of 2.84, outperforming the non-prompted condition ($d = 2.15$). The difference between the SP group ($d = 2.37$) and the GP group ($d = 3.30$) is also prominent.

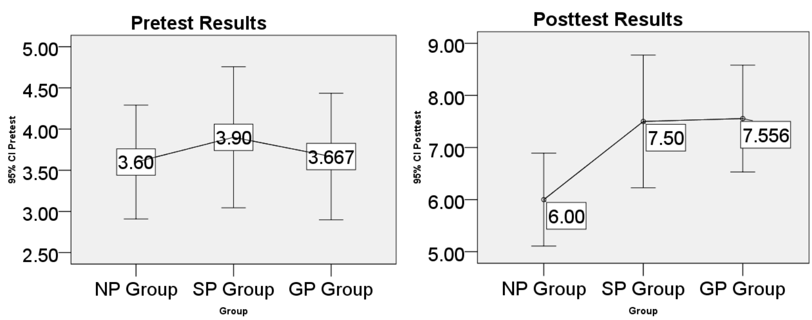


Fig. 15.5 Pretest and Posttest Scores

Table 15.3 Result of Knowledge Pre-/Post-Test

Groups	N	Pre-test (Mean/SD)	SE Post-test (Mean/SD)	Cohen's *d*
NP Group	10	3.60 (0.97)	6.00 (1.25)	2.15
SP Group	10	3.90 (1.20)	7.50 (1.78)	2.37
GP Group	9	3.67 (1.00)	7.56 (1.33)	3.30

15.4.3 Impact of Question Prompts on Self-Efficacy Development

To test students' development of self-efficacy, we compared the students' pre-to-post scores in the pre-test on self-efficacy. The data is reported in Table 15.4. The effect size computed as Cohen's d suggests that, although most groups experienced little progress in self-efficacy during such a relative short-term experimental study, the GP Low group has experienced a prominent progress ($d = 6.27$) than others. A Tukey's test, performed to compare the difference between groups, also reveals that there is a significant difference between the GP Low group and the Control Low group ($MD = 21.75$, $p = 0.003$). It suggests that generic prompts could support students with low self-efficacy to develop their level of self-efficacy through the learning activities.

15.4.4 Impact of Self-Efficacy on Adoption of Question Prompts

Table 15.5 shows the result of domain knowledge pretest to posttest. It shows the effect size (Cohen's d) of each group from knowledge pretest to posttest. Students evaluated as having high efficacy receiving generic prompts acquired the most progress (Cohen's $d = 4.17$). Such a result implies that students with high efficacy, who received generic prompts could achieve better learning outcomes than other groups of students.

Table 15.4 Result of Knowledge Pre-/Post- Test

Groups	N	SE Pre-test (Mean/SD)	SE Post-test (Mean/SD)	Cohen's *d*
Control Low	5	23.00 (10.36)	27.00 (10.36)	0.39
Control High	5	43.89 (7.08)	45.00 (8.00)	0.15
GP Low	4	20.50 (4.20)	48.75 (4.78)	6.27
GP High	5	47.00 (7.00)	49.40 (4.87)	0.40
SP Low	5	25.40 (8.93)	35.40 (8.82)	1.13
SP High	5	45.20 (8.29)	43.40 (6.07)	-0.24

Table 15.5 Result of Domain Knowledge Pre-/Post- Test

Groups	N	Pre-test (Mean/SD)	Post-test (Mean/SD)	Cohen's *d*
GP High	5	4.25 (0.96)	8.25 (0.96)	4.17
GP Low	4	3.20 (0.84)	7.00 (1.41)	3.27
SP High	5	4.00 (1.58)	7.60 (2.30)	1.82
SP Low	5	3.80 (0.83)	7.40 (1.34)	3.22

15.4.5 Impact of Question Prompts on Elicited Reflection Levels

A tripartite coding scheme, adapted by Ortiz (2006) from the categorization scheme proposed by Surbeck et al. (1991), was adopted to analyze the participant' response statements to the agent prompts. This scheme, includes categories of reaction, elaboration, and contemplation, from lowest to highest, adopts the perspective that the nature of the stimulus to reflect will impact the quality of the reflection (Moore and Whitfield 2008), which conforms to what this work study.

An initial observation of the transcript excerpts of prompt-response exchange showed that the GP participants exhibited a higher level of reflection than the SP participants. We noted that the GP participants gradually tended to present more completive response statements than the SP participants after being exposed to the question prompts for a period of time. A sample transcript of participants' response statement is outlined as follows to demonstrate both SP participant and GP participant responded to the purposefully designed same question prompts but showed different levels of reflection in their response statements.

[Simulated Tutee] What is the key thing you try to teach me?
[SP Participant] What is economics? (Reaction: Report facts)
[GP Participant] The universality of these links, causes and implications. (*Contemplation: Indicate a shift in thought*)
[Simulated Tutee] What is the key thing you learn from me?
[SP Participant] Demand, supply and price are interrelated and contribute to the idea of economics in total. (*Elaboration: Provide interpretive analysis of course concepts*)
[GP Participant] What to do when approaching economics in real life. (*Contemplation: Indicate a shift in attitude about himself*)
[Simulated Tutee] Do you learn something from me?
[SP Participant] Yes, I learn how to organize my thoughts better. (*Contemplation: Indicate a shift in attitude about himself*)
[GP Participant] How to teach better and more clearly. (*Contemplation: Indicate a shift in attitude about himself*).

Two raters were involved in the analysis of response statements. To begin, the first rater (the first author) analyzed a sample of ten response statements to become confident with the coding scheme. Then, the first rater explained the coding scheme to the second rater (a researcher specialized in educational research).

The second rater analyzed the first ten response statements as well and the differences between the two raters were discussed to differentiate between *reaction*, *elaboration,* and *contemplation*.

After that, the two raters analyzed the remaining response statements independently. Cohen's kappa was used to estimate the level of agreement between the 2 raters, taking the agreement occurring by chance into account. Calculated with Cohen's Kappa, the agreement between the two raters appeared to be good, $\kappa = 0.653$ (A Cohen's kappa value between 0-0.4 is considered poor, 0.4-0.6 fair, 0.6-0.75 good, and 0.75 outstanding).

A combined qualitative and quantitative analysis of participants' response statements to simulated tutees' question prompts (a total number of 286 response statements were analyzed) showed the difference in the levels of reflection between groups (Table 15.6). An ANOVA test shows a statistical significant difference between the groups as to reactive statements ($F\ (1, 17) = 36.747$, $p <.05$) and contemplative statements ($F\ (1, 17) =19.472$, $p < .05$). The number of elaborative statements was not significantly different between the groups. Such a result shows that the participant of GP group, whether with high or low efficacy, was more likely to respond with contemplative statements representing a higher level of reflection. Comparatively, the participants of SP group, whether with high or low efficacy, responded more with reactive statements representing a lower level of reflection which means they pay more attention to report issues with no development than the GP group.

15.5 Conclusions

The results suggest that the use of agent tutee as an active and inquisitive learning partner to raise meaningful questions could be helpful to students learning by reflective teaching. The preliminary results, quantitative and qualitative, suggest that the generation of agent prompts as computer-based scaffolds, when adapted to student learning-by-teaching activities, could be useful in particular ways, such as improving students' learning outcomes and eliciting higher levels of reflection.

Table 15.6 Result of Response Statements Analysis

Levels of Reflection	GP (Mean/SD)	SP (Mean/SD)	ANOVA-Test[1]
Reaction	6.00 (1.41)	9.60(1.17)	36.75*
Elaboration	7.56 (1.54)	9.50(2.17)	4.19
Contemplation	9.00 (1.80)	5.10 (2.82)	19.47*

*, p < .05

Specially, this study suggests that generic prompts could possibly be beneficial in the development of self-efficacy, thus, leading to fostering self-directed learning. However, there remain significant challenges that must be overcome before a system similar to the one used in this study, with agent prompts, can be incorporated into regular classroom use. Generally, the goal of supporting students to learn needs a shift from a notion of leading to one of facilitating and enabling. This means designing agent tutee systems that are not necessarily recipients of information but, rather, a facilitator to promote reflection. Instead of attempting to create agent tutee systems that will always know the correct answer, designers need to invent agent tutee systems that encourage students to do both reflection-in-action and reflection-on-action (Schön 1987) and attend more on double-loop reflection (i.e., think out of the box) (Argyris and Schön 1996).

A future research direction might be to investigate the design of current learning-by-teaching system to involve intelligent mixes of generic prompts and specific prompts. For example, an intelligent mix might give students specific prompts for their first experiences with prompting, and then fade into generic prompts.

An alternative improvement of system design would be tailoring the prompts given on the basis of individual student characteristics like self-efficacy. Students with high efficacy might be given the generic prompts, whereas, students with low efficacy might receive the intelligent mix of generic and specific prompts.

Possible future work also includes exploring the relationship between agent prompts and flow. Flow is a state of the mind, individuals reached that made them very productive in the task that they perform (Csikszentmihalyi 1978). As Csikszentmihalyi notes, persons in flow feel completely involved in, and focused on their task because of their curiosity or training. They feel great inner clarity, know that the activity is doable, do not notice time passing, and feel intrinsically motivated. Flow is a highly activated state of the mind that puts the person performing a task in control. Putting a learner in flow some of the time would be a highly appreciable goal for learning environments (Katzlberger 2005).

However, agent prompts inherently interrupts the learner's learning tasks at-hand. To achieve flow, one key issue is to facilitate incorporating and fading agent prompts for the students at the right moment. More refined approaches are needed to analyze, when the students are embarking on a new activity and when they have finished that activity and are almost ready to move on to the next steps. The answers will provide an in-depth understanding of intellectual enjoyment that students encounter, when using agent tutee systems as learning partners.

Acknowledgments. We thank Dr. Gautam Biswas in Vanderbilt for providing the Teachable Agent software.

References

Aleven, V., Koedinger, K.R.: An effective metacognitive strategy: learning by doing and explaining with a computer-based cognitive tutor. Cognitive Science 26, 147–179 (2002)

Aleven, V., Pinkwart, N., Ashley, K., Lynch, C.: Supporting self-explanation of argument transcripts: specific v. generic prompts. In: Proceedings of ITS Workshop of Intelligent Tutoring Systems for Ill-Defined Domains, pp. 47–55 (2006)

Amulya, J.: What is Reflective Practice? (2004), http://www.itslifejimbutnotasweknowit.org.uk/files/whatisreflectivepractice.pdf (accessed April 18, 2009)

Argyris, C., Schön, D.: Organizational learning II: Theory, method and practice. FT Press, Mass (1996)

Azevedo, R., Hadwin, A.F.: Scaffolding self-regulated learning and metacognition-Implications for the design of computer-based scaffolds. Instructional Science 33, 367–379 (2005)

Biswas, G., Schwartz, D., Bransford, J. (TAG-V): Technology support for complex problem solving: From SAD Environments to AI. In: Feltovich, F. (ed.) Smart Machines in Education, pp. 71–98. AAAI Press, Menlo Park (2001)

Brockbank, A., McGill, I.: Facilitating reflective learning in higher education. Society for Research into Higher Education. Open University Press, Buckingham (1998)

Chang, K.-E., Sung, Y.-T., Chang, R.-B., Lin, S.-C.: A new assessment for computer-based concept mapping. Educational Technology & Society 8(3), 138–148 (2005)

Chen, N.S., Kinshuk, Wei, C.W., Liu, C.C.: Effects of matching teaching strategy to thinking style on learner's quality of reflection in an online learning environment. Computers & Education 56(1), 53–64 (2011)

Chen, N.S., Wei, C.W., Wu, K.T., Uden, L.: Effects of high level prompts and peer assessment on online learners' reflection levels. Computers & Education 52(2), 283–291 (2009)

Chen, S.W., Lin, S.C., Chang, K.E.: Attributed concept maps: fuzzy integration and fuzzy matching. IEEE Transactions on Systems, Man, And Cybernetics 31(5) (2001)

Chi, M.T.H., de Leeuw, N., Chiu, M., LaVancher, C.: Eliciting self-explanations improves understanding. Cognitive Science 18, 39–477 (1994)

Cohen, J.: Theoretical considerations of peer tutoring. Psychology in the Schools 23, 175–186 (1986)

Cohen, P.A., Kulik, J.A., Kulik, C.C.: Educational outcomes of tutoring: A meta-analysis of findings. American Educational Research Journal 19(2), 237–248 (1982)

Coleman, E.B., Brown, A.L., Rivkin, I.D.: The effect of instructional explanations on formal learning from scientific texts. The Journal of the Learning Sciences 6(4), 347–365 (1997)

Cornford, I.R.: Learning-to-learn strategies as a basis for effective lifelong learning. International Journal of Lifelong Education 21, 57–368 (2002)

Csikszentmihalyi, M.: Intrinsic rewards and emergent motivation. In: Lepper, M.R., Greene, D. (eds.) The Hidden Costs of Reward. Lawrence Erlbaum Associates, Hillsdale (1978)

Davis, E.A.: Scaffolding students' reflection for science learning. PhD Thesis. University of California, Berkeley, CA (1998)

Davis, E.A.: Prompting middle school science students for productive reflection: Generic and directed Prompts. The Journal of The Learning Sciences 12(1), 91–142 (2003)

Davis, E.A., Linn, M.: Scaffolding students' knowledge integration: Prompts for reflection in KIE. International Journal of Science Education 22(8), 819–837 (2000)

Gama, C.: Metacognition in Interactive Learning Environments: The Reflection Assistant Model. In: Lester, J.C., Vicari, R.M., Paraguaçu, F. (eds.) ITS 2004. LNCS, vol. 3220, pp. 668–677. Springer, Heidelberg (2004)

Gartner, A., Kohler, M., Riessman, F.: Children teach children: Learning by teaching. Harper & Row, New York (1971)

Ge, X., Land, S.M.: A conceptual framework of scaffolding ill-structured problem solving processes using question prompts and peer interactions. Educational Technology Research and Development 52(2), 5–27 (2004)

Gouli, E., Gogoulou, A., Papanikolaou, K.A., Grigoriadou, M.: An adaptive feedback framework to support reflection, guiding and tutoring. In: Magoulas, G., Chen, S. (eds.) Advances in Web-based Education: Personalized Learning Environments, pp. 178–202. Information Science Publishing, New York (2005)

Graesser, A.C., Person, N.K., Magliano, J.P.: Collaborative dialogue patterns in naturalistic one-to-one Tutoring. Applied Cognitive Psychology 9, 495–522 (1995)

Graesser, A.C., VanLehn, K., Rose, C., Jordan, P., Harter, D.: Intelligent tutoring systems with conversational dialogue. AI Magazine 22, 39–51 (2001)

Hmelo, C., Day, R.: Contextualized questioning to scaffold learning from simulations. Computers & Education 32, 151–164 (1999)

Katz, S., O'Donnell, G., Kay, H.: An approach to analyzing the role and structure of reflective dialogue. International Journal of Artificial Intelligence and Education 11, 320–333 (2000)

Katzlberger, T.: Learning by teaching agents. PhD Thesis, Vanderbilt University (2005)

King, A., Staffieri, A., Adelgais, A.: Mutual peer tutoring: Effects of structuring tutorial interaction to scaffold peer learning. Journal of Educational Psychology 90(1), 134–152 (1998)

Kornilakis, H., Grigoriadou, M., Papanikolaou, K.A., Gouli, E.: Using WordNet to support interactive concept map construction. In: ICALT (2004)

Kuhn, D., Udell, W.: The development of argument skills. Child Development 74(5), 1245–1260 (2003)

Leelawong, K.: Using the learning-by-teaching paradigm to design intelligent learning environments. PhD Thesis, Vanderbilt University (2005)

Mason, B., Bruning, R.: Providing feedback in computer-based instruction: What the research tells us (2003), http://dwb.unl.edu/Edit/MB/MasonBruning.html

Moon, J.: A Handbook of reflective and experiential learning. Routledge, London (2004)

Moore, J., Whitfield, V.F.: Musing: A way to inform and inpire pedagogy throught self-reflection. The Reading Teacher 61(7), 586–588 (2008)

Mory, E.: Feedback research. In: Jonassen, D.H. (ed.) Handbook of Research for Educational Communications and Technology, pp. 919–956. Simon & Schuster Maxmillan, New York (1996)

Ortiz, J.: Reflective practice and student learning in the introductory interpersonal communication course. Technical report, Maricopa Institute for Learning (2006)

Peverly, S.T., Wood, R.: The effects of adjunct questions and feedback on improving the reading comprehension skills of learning-disabled adolescents. Contemporary Educational Psychology 26(1), 25–43 (2001)

Pintrich, P.A., Smith, D.A.F., Garcia, T., McKeachie, W.J.: A manual for the use of the motivated strategies for learning questionnaire. Technical report, The University of Michigan (1993)

Roscoe, R.D., Chi, M.T.H.: The influence of the tutee in learning by peer tutoring. In: Forbus, K., Gentner, D., Regier, T. (eds.) Proceedings of AMCSS, Chicago, pp. 1179–1184 (2004)

Roscoe, R.D., Chi, M.T.H.: Understanding tutor learning: Knowledge-building and knowledge-telling in peer tutors' explanations and questions. Review of Educational Research 77(4), 534–574 (2007)

Rothkopf, E.: Learning from written instructive materials: An exploration of the control of inspection by test-like events. American Educational Research Journal 3, 241–249 (1966)

Sandoval, W.: Conceptual and epistemic aspects of students´ scientific explanations. The Journal of the Learning Sciences 12(1), 5–51 (2003)

Schön, D.A.: Teaching artistry through reflection-in-action. Educating the Reflective Practitioner. Jossey-Bass Publishers, San Francisco (1987)

Schraw, G.: Promoting general metacognitive awareness. Instructional Science 26, 113–125 (1998)

Surbeck, E., Han, E.P., Moyer, J.E.: Assessing reflective responses in journals. Educational Leadership 48(6), 25–27 (1991)

VanLehn, K., Jones, R.M., Chi, M.T.H.: A model of the self- explanation effect. Journal of the Learning Sciences 2(1), 1–60 (1992)

Weinstein, C.E., Meyer, D.K.: Cognitive learning strategies and college teaching. New Directions for Teaching and Learning 45, 15–26 (1991)

Xie, K., Bradshaw, A.C.: Using question prompts to support ill-structured problem solving in online peer collaborations. International Journal of Technology in Teaching and Learning 4(2), 148–165 (2008)

Zimmerman, B.J.: Self-Efficacy: An essential motive to learn. Contemporary Educational Psychology 25(1), 82–91 (2000)

Abbreviations

ANOVA	Analysis of Variance
ILE	Intelligent Learning Environment
ITS	Intelligent Tutoring System
GP	Generic Prompts
MD	Mean Difference
MSLQ	Motivated Strategies for Learning Questionnaire
SP	Specific Prompts

Chapter 16
Acquisition of Higher Order Knowledge by a Dynamic Modeling Environment Based on the Educational Concept of Self-Regulated Learning

Stefanie A. Hillen

University of Agder
Gimlemoen 25, Kristiansand, Norway
Stefanie.a.hillen@uia.no

Abstract. I aim to show that learning with this modeling based Educational Learning System (ELS) can accomplish the target of achieving higher order knowledge. The ELS is a system consisting of internal and external elements. The external prerequisites consist of technical and physical elements and the internal ones are shaped by the students pre-knowledge and the instructors teaching competencies including his/her social, emotional, and disciplinary knowledge necessary for teaching. The ELS is based on a theoretical framework of different theories and models such as concept mapping, elaboration of mental models, cognitive tool-approach, and self-regulated learning (SRL). Different features for visualization and modeling of the subject matter to be learned can be chosen by the students as well as the frequencies using the simulation feature to receive feedback to the model constructed. This enables the students to work self-regulated because of the feedback of the system, by providing the simulation results in desired graphical or analytical representation formats. The notation of the ELS, the symbols themselves are considered as an intuitive language because the symbols are connected to real world phenomena. It is assumed that the expression of knowledge is co-determined by the applied language. It is concluded that a less differentiated language does not hinder thinking but does not support thinking as a 'cognitive tool'. Hence the hypothesis is: there are significant differences in the complexity of the expressed knowledge of the students using the notation in comparison to a control group using verbal protocols to express the knowledge acquired.

16.1 Introduction

The goal of the ELS by using the system dynamics (SD) approach (Forrester 1961) is to allow students to grasp complexity in the dimensions of magnitude and dynamics (Ulrich and Probst 1995).

A. Peña-Ayala (Ed.): Intelligent and Adaptive ELS, SIST 17, pp. 393–416.
springerlink.com

Complexity is often understood as a big number of aspects and how they are related to each other (Dörner 1976) which makes thinking and acting complicated. The fact that systems are changing over time (dynamics) is somehow not stressed or even more not taught in education. The learning effectiveness of the SD-based ELS is shown by the learning outcomes of students through the means of different knowledge representation formats. Knowledge representation formats or notations as well as language are shaping through their grammar - situations, images, and procedures - in a specific context (Dörner 1976). Dörner stresses that: "Language may have an impact on the processes of thinking". Oerter 1971 assumes interdependence between the way of thinking and the language used; that is, the depiction of knowledge is co-determined by the applied language. On consistently following the ideas of Oerter and Dörner, a differentiated language with grammar and rules eases the opportunity to express thoughts, because the structures for externalizing thoughts are given and haven't to be invented again. It is consequent to conclude that a less differentiated language does not hinder thinking but does not support thinking as a 'cognitive tool'.

This externalization by the system dynamics based ELS can be understood as providing feedback to the students during their learning processes. Even the ELS can be described as intelligent and adaptive it cannot be used as auto-didactical or as a purely self-directed learning application. The external pre-requisites consist of technical and physical elements and the internal ones are shaped by the students prerequisites and pre-knowledge as well as the instructors teaching competencies including his/her social, emotional, and disciplinary knowledge necessary for teaching.

Studies on computer based learning activities often analyzing deeply the processes of modeling as a learning activity. Often the learning outcomes or learning effectiveness is on less close examinations. Van Borkulo et al. 2009 conducted a literature review on several studies on modeling and simulation confirming this situation. One reason might lie in the difficulty to assess learning processes as well as learning results on higher order knowledge. The research group (Spector et al. 2001) found out that researchers in this field think that a conventional measurement is not appropriate for such ELS focusing on higher order knowledge of complex domains. Hence, this study is focusing on the representation of learning processes and on the assessment on higher order knowledge. It traces the modeling activities and analyzes the learning products for assessment. In addition, a diagnostic tool for measurement of the learning outcomes (explicates) was developed.

16.2 The Theoretical Framework

The ELS is based on a theoretical framework of different theories and models such as concept mapping (Ryssel et al. 2008), elaboration of mental models (Johnson-Laird 1988), cognitive tool-approach (Jonassen 1991) and self-regulated learning (Bandura 1997).

Those theoretical approaches shaped the use of technology and applied for teaching and learning purposes. A misconception in the application of technology for learning is unfortunately observable – there is an assumption that applying a new technology is already a learning approach. This does not exclude that learning can take place or furthermore leads to complex learning results but the teaching intension can hardly be stressed. This approach makes use of an instructional approach by integrating external and internal pre-requisites for an educational learning environment based on the theoretical framework explained below.

16.2.1 Concept Mapping

In educational and psychological contexts, the representation of knowledge and knowledge structures by concept mapping are targeting on visualization and accessibility. The representation over time enables to observe the learning progress. Nevertheless, mapping tools, techniques and methods to represent knowledge are more than just the visualization or assessment of the knowledge-base of learners (Ryssel et al. 2008). It is assumed that the process of depicting knowledge is a knowledge constructing process itself. This approach is not new or not one of the results of computer based education. It can also be found in Scheele and Groeben 1988 as a paper and pencil approach, structure laying technique, where subjective theories should be made visible.

Concept maps represent the expressed structural knowledge of a learner by the time (static concept map). These structural representations consist of concepts which are connected by formal relations. During the process of (re)construction of subjective theories guided by a moderator, partly ‘new’- knowledge is constructed or generated. This procedure has an inherent dynamic for knowledge construction. The revealed knowledge made visible and assessable through file cards is even more developed and specified through the structural (re)organization of the file cards by the learner (Dann 1992, Scheele and Groeben 1988). Another reference can be seen to semantic networks (Collins and Quillian 1969). If a ‘static’ concept mapping approach such as described in Sect. 16.2.1 can support learning, than this can be analogically concluded for the dynamic representation of mental models too (Johnson-Laird 1983, 1988). The question of external representation for enhancing learning, in problem solving is also stressed by Kolloffel et al. 2010.

16.2.2 Elaboration of Mental Models

The hypothetical construct of mental models is not consistently defined. Nevertheless there is some conformance. It is stressed that a mental model has to be functional, and that such functionality corresponds to one of is its main characteristics (Johnson-Laird 1983, Norman 1983).

However, an interesting statement about mental models is the one provided by Norman 1986:”Mental models seem a pervasive property of humans. I believe that people form internal, mental models of themselves and of the things and people with whom they interact. These models provide predictive and explanatory power for understanding the interaction. Mental models evolve naturally in interaction with the world and with particular systems under consideration. These models are highly effected by the nature of the interaction, coupled with the person’s prior knowledge and understanding. The models are neither complete nor accurate, but nonetheless they function to guide much human behavior.”

Mental models are functional because they support the understanding of situations, guide and define cognitive, and physical operations (Kluwe and Haider 1990). Mental models are incomplete, stable, parsimonious, and unscientific (Norman 1983). Astonishing it is a somehow obviously wrong applied theory (i.e. mental model of a thermostat versus a valve) in everyday-life it is sometimes as well as viable as the appropriate theory. This leads to the stability of naïve mental models (Mandl and Spada 1988). Naïve Mental Models are set by visualization. Through SD-models improve the knowledge base of students. The ELS presented is based on explorative and empirical research that mental models are outcome by the SD-based simulation and modeling activities (Hillen et al. 2000, Hillen 2004).

16.2.3 Cognitive Tool Approach

Cognitive tools can support cognitive and meta-cognitive processes if cognitive tools are applied appropriately. Cognitive tools are construction tools of knowledge, which extend the reach of mind (Salomon 1988). An aim is placed on the interrelation between stating thoughts and the cognitive tool by Bliss 1994: “Written language, drawing, paintings... are some of our ways to externalizing thinking. They have the virtue of permanency, so becoming available to reflect on. Computer modeling may be able to help students to think.”

A cognitive tool can disburden from routines, which would otherwise, hinder to reach and manage the ‘higher order thinking’ procedures. This can be explained by the cognitive load theory (Sweller 1988). His theory discussed the distribution and application of cognitive resources. Nevertheless, cognitive tools should not ease or simplify thinking. The intension is not to apply cognitive tools as a fingertip tool like it is known from a calculator but instead allow learners to accomplish thinking procedures and processes which would not have been possible without the tool. Reducing cognitive efforts will lead to a less mindful elaboration of the subject matter. Jonassen 1991 mentioned: “Rather cognitive tools provide an environment and vehicle that often requires learners to think harder about the subject matter domain being studied while generating thoughts that would be difficult without the tool. They are cognitive reflection and amplification tools that help learners to construct their own realities using the constructs and processes in the environment on a new content domain.”

Unfortunately cognitive tools are neither stand-alone tools nor automatically effective. Salomon termed it as an intellectual partnership (Salomon et al. 1991). A useful application and an embedding in a meaningful learning-environment has to be designed and developed. In addition, handling routines with the tool have to be learned or have to be integrated or supported by the learning environment, which is task of instructional design. Beside an appropriate design of the learning environment a learner's prerequisite for this intellectual partnership is the student's ability of SRL. This can be concluded because the use of cognitive tools are based on i.e. the learner's (cognitive) reflection.

16.2.4 Self-Regulated Learning

Zimmerman and Schunk 2001 question the concept of SRL as: "oxymoron". On the one hand the word *regulation* refers to keeping something regular; whereas *learning* refers to changes in performance produced by experience. They solved it by mentioning that SRL seeks to explain how people improve their performance using a systematic or regular method of learning. Self-regulation has an important impact on motivation and performance in learning (Bandura 1997) and it should be acknowledged for learning purposes. The ability of self-regulation is known as a prerequisite for learning as well a result by it. Metacognitive activities are monitoring and controlling the act of learning. Zimmerman 1989 states: "Students can be described as self-regulated to the degree that they are metacognitively, motivationally, and behaviorally active participants in their own learning process."

There is an agreement between researchers about the prerequisites for SRL (see Simons 1992); Firstly, learning has to be prepared (pre-knowledge has to be activated). Secondly, the act of learning has to be conducted and finally learning has to be regulated by the use of control- and evaluation strategies to maintain the motivation and the concentration on a learning task or a learning procedure. Planning, organizing and evaluations are the typical characteristics of a self-regulated learner. One assumption is that the goal of the self-regulated learners is to become their own teachers; that is, to shift from being taught to merely self-reflective practice (Schunk and Zimmerman 1998). This does not exclude the responsibility of the teacher to offer chances for SRL by either providing an advance organizer[1] (Ausubel 1960) or a well elaborated didactical structure of the lesson. A concept map can be seen as an advance organizer itself (Willerman and MacHarg 1991).

A number of researchers suggest that one potential mediator of computer-based learning environments and academic performance is the quality of student's self-regulatory learning (Winters et al. 2008). Their study revealed that even if there are different theories on SRL mostly all of these are characterized by a number of phases following in a loose order.

[1] An advance organizer provides support for the knowledge acquisition. This is set by directing attention to what is key in the lesson and by stressing the relation to the student's pre-knowledge.

The phases can be described as follows: Planning and forethought signify the first phase. The following step is the conduction of the plans and strategies. In this phase metacognitive and monitoring strategies are applied to control the learning process. The last phase contains reflective activities on the former learning experiences made (e.g., the performance is evaluated). This reflection leads to a revision or adaption of the individuals' self-beliefs, beliefs about tactics, strategies and learning context which influence learning activities. Self-regulation skills are seen as influencing variable or as mediator for motivation and performance not just for computer-based learning environments but as well as potentially decisive for gaming approaches (Hillen et al. 2011).

Vandevelde et al. 2011 report the impact of students tutoring which attests that: "Self-regulation seen as cross curricular skills can be increased by teaching." Nevertheless they are advocating as well the need for more comprehensive research in this research area.

16.3 ELS: The Application of Technology for Learning Purposes

Like discussed before, the process of knowledge acquisition is intended to be supported by the SD-based learning environment (ELS) through the feature of visualization and (re)construction by modeling and simulation. It can be assumed that this kind of feature provides opportunities for the application of metacognitive activities as well as it asks for metacognitive activities. As stressed above the ability of self-regulation is a prerequisite as well as a result of learning.

16.3.1 Feedback

The feedback to students is given in two different ways: 1) conceptual by the congruency to the subject matter 'business administration' of the case provided for instance the 'demand function'; 2) technological functional (e.g. equations and units) by the system itself (see Fig. 16.1; 16.2). It has to be admitted that the differentiation between a technological and conceptual evaluation can just be done analytically but it is not possible related to the appropriateness of the mental model constructed by the students. Having constructed a preliminary model (see Fig. 16.1) the conceptual and technological functionality of the model can be evaluated by its simulation.

The simulation feature can be applied using behavior over time graphs, tables, or icons representing the actual levels of the variables or all options at once. These visualizations features can be chosen by the students themselves as well as the frequencies using the simulation feature to receive feedback to the model constructed.

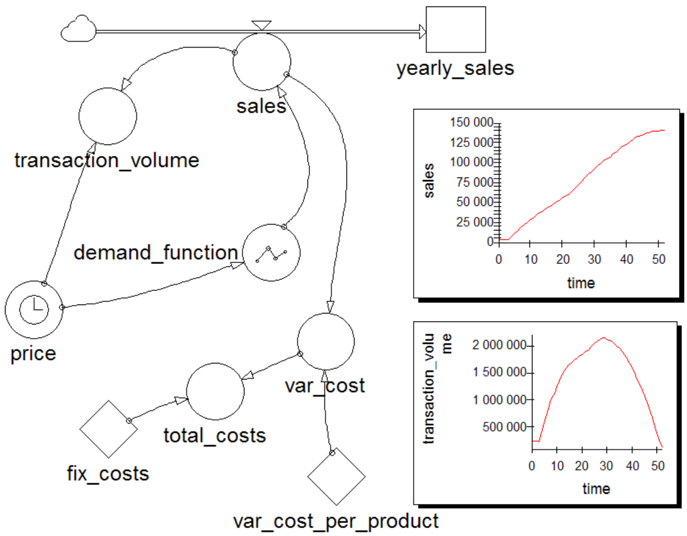


Fig. 16.1 A student constructed model of the concept ‘price, demand function, and sales`

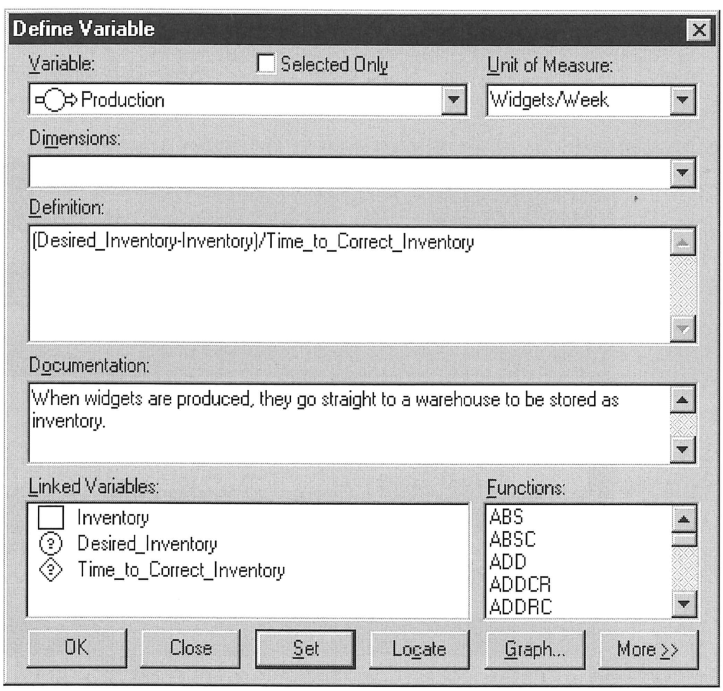


Fig. 16.2 Definition window as visualization for the appropriate definition, documentation and the linkage of variables of the model (screenshot)

This enables the students to work self-regulated because of the feedback of the system, by providing the simulation results in desired graphical or analytical representation format. Comparing the given case on the worksheet (see Sect. 16.4.2.1), the assumption of the behavior and the actual results provided by the simulation show how 'functional' students' assumptions are (Hillen 2011).

The instructional guidelines asking (by the worksheet) for saving the states of model, which means to freeze it by the student themselves to make sure that the construction process of model become visible as well as to avoid not reflected modifications, like trial and error activities. Remodeling and simulation import the danger of missing the conscious process of knowledge construction. Being aware of what has been learnt is one part of SRL; which is metacognition. To freeze the model helps the learner starting rethinking and reflecting on the simulation and the simulation results. By those 'frozen' model states the learner, the teacher or the whole class is able to reflect and evaluate on the model structure, content and behavior. By this the students are fostered to verify these results. In the worst case the simulation cannot be conducted if there are mistakes in the model (e.g. by a wrong definition of a variable or unit which represent the concept too). The system offer the option to mark undefined or wrongly defined variables with question marks (see Fig. 16.2; 16.4). The description of the rate-variable 'production' is the documentation of the concept beside the formal definition in the variable window.

The definition reveals the division of three linked variables, the 'Desired_Inventory-Inventory' and 'Time_to_Correct_ Inventory'. The window of linked variables shows that two variables are not properly defined.

Interactivity (e.g. control) and feedback are not divided in single features. The interplay of the features used by the self-regulated learner led to the individual chosen feedback of the system (see Fig. 16.5). The feedback of the system is of course based on the technical framework but the feedbacks do as good as the users, the teacher's expertise as well as the student's capability to use it. This does not imply that the student has to become knowledgeable about the technical subtlety. It is more decisive to have an educational setting which makes use of the features appropriately. This starts with useful cases. Thus a collection of cases has been developed for teachers including worksheets and guidelines.

16.3.2 The Reference System and the Modeling Curriculum

The reference system is not based on one or several student models. There are a 'bunch' of models given by the reference system in relation to the domain of business administration. Like mentioned above the students' progress is traced by the modeling steps. This tracing is done periodically not related to time but stepwise to the task performed by the students' incrementally improved model (see Fig. 16.1). As a technical reference, a generic model of an SD-based industrial enterprise model was designed (Molkenthin et al. 2008). This enterprise model is targeting on the measurement of the performance of control by the students.

The representation of an industrial enterprise is congruent to the curriculum in business administration (Beschluss der KMK 2002). Different activities of such an enterprise, production, were taken as basis of cases students were asked to build.

Because the business world is individual, neither enterprise behavior nor any organizational structure will look alike. Therefore, the congruencies to the generic business concepts were used as reference system and not single individual pre-designed models. Furthermore, as the reference system is a curriculum on business administration concepts specifically on office clerks , it asked for " … the students should be able to detect, recognize, discover, and to describe the organization as well as the business processes …" (Beschluss der KMK 1995). The support of the acquisition of knowledge to solve complex problems through exemplary tasks is a part of the VET-program (Beschluss der KMK 1995). Such is also applied in other subjects under VET school (e.g., natural sciences) (PZ-Informationen 2000). The idea to use the curriculum as a reference system is also applied in other modeling and simulation activities. The term '*modeling curriculum*' mirrors this idea (Alessi 2009). The intension with the modeling curriculum is based on the principle that learners will better understand a system or a phenomenon by developing and refining a working computer model of that system or that phenomenon.

16.3.3 The SD-Based ELS to Support SRL

SD was outlined at MIT in 1956 (Sterman 2000). It combines theory, methods, and philosophy needed to analyze the behavior of systems. Various technical applications exits which are based on the same idea on SD.

There are several tools using modeling and simulation (de Jong 2010) e.g. the Co-Lab modeling tool (van Joolingen et al. 2005). With the modeling tool students represent the ideas about the domain, applying the variables and its relations. Those constructed models can be simulated to compute the values of the variables over time. This enables the learner to observe the course of the events predicted by the variables. The student compares the calculated data with the data predicated. Thus the student may accept the model, adapt or revise it again. Co-Lab follows the idea which is presented below but this SD-based ELS tries to stress specifically SRL by the internal and external requisites of the learning environment.

If applying technology and knowing that self-regulation cannot be assumed as given or as sufficiently pre-existent by the learner one question emerges: How to support, enable, and foster such metacognitive/self-regulating activities by a technology based learning environment?

If one looks the activities required and supported by the SD-based ELS and the following three activities (Simons 1992) it seems to match each other:

1. The activation of pre-knowledge can be supported by the visualization of the naïve mental model by the SD-based ELS initialized by the curriculum based case,

2. The worksheet structure supports the elaboration of the mental model asking critical questions to the model behavior and causes hereby cognitive conflicts,
3. It leads to further control and evaluation activities of the student that are demanded by the simulation and interpretation or explication of the results.

The decisive items and features of the SD-application are the notation, the organization and simulation functions as well as the possibility of creating user interfaces for educational purposes. The use of these features contributes to SRL and support the acquisition of higher order thinking considering an underlying pedagogy. To learn a combination of both technology and pedagogic is needed.

16.3.3.1 Cyclic Steps

The learning approach can be divided into iterative but cyclic steps. A similar pedagogical approach how to teach naïve learners is suggested (see Alessi 2009). Similar to Alessi's work van Borkulo et al. 2009 term it: "The ACE framework". That framework comprises applying a model, creating a model, and evaluating a model.

The cyclic steps in this study are: 1) conceptualization of the naïve mental model; 2) visualization and specification of the concepts by applying the notation on the screen; 3) re-construction, simulation, and reflection for further work.

Concerning the conceptualization and application of the notation, the SD-notation consists of different single symbols and is not a programming language to learn. Nevertheless the users have to specify the concept with its interrelation to other variables (see Fig. 16.2). A new challenge to the user is the organization of the symbols of the notation. If the variables do not fit because of the notation's logic the user receive a feedback in form of pop up-messages.

As regards with the reorganization and simulation, the organization of the symbols reflects the content of the subject matter's structure, which has to become shaped in a functional form through the connection of the single symbols. The more complex a concept is the more variables and interrelation will the dynamic concept map have. The symbols are considered as intuitive because they are connected to real world phenomena: a flow (a faucet), a level (a bathtub to be filled or emptied), and an information link (checking the water level) (see Fig. 16.3).

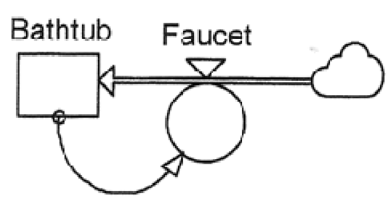


Fig. 16.3 The logic of the systems' notation

It is assumed that the elaboration of knowledge can be supported by its organization and structuring. In specific, higher order knowledge including behavior over time (e.g. exponential development) and the magnitude of the subject matter can be represented by the notation. This argumentation can also be found in studies as an opportunity for supporting scientific reasoning (Borkulo et al. 2008).

Simulation and reflection for further work on the model: The simulation can be made visible by tables, graphs over time as well as by animated symbols in the notation itself, that is, there are several visualization tool presenting the process and the results of a simulation (see Fig. 16.5; 16.2). The simulation is mathematically seen as a continuous simulation approach[2]. The simulation runs can be automated; parameters can be changed by gauges before and during the simulation. The simulation feature supports the cognitive simulation, that is, the elaboration of the mental model. In addition, the visualization helps to reduce the cognitive load during problem solving (Sweller 1988).

16.3.3.2 Features and Restrictions to Support the Self-Regulated Learner

Like explained above to support the student's self-regulation activities the learner can choose what kind of visualizations he or she wants (see Fig. 16.4; Fig. 16.5). Winters et al. report out of their study that it is often up to learners to determine which representation are most useful, based on their self-knowledge and beliefs, motivational factors, prior knowledge, task definitions, goals, and strategic knowledge (Winters et al. 2008).

This careful considered engagement is indicative for a self-regulated learner (Zimmerman and Schunk 2001, Zimmerman 2008). Nevertheless, empirical research (Lajoie and Azevedo 2006) has shown that students often struggle when using computer based learning environments providing multiple representations of information and opportunities to manipulate them.

While exploratory learning advocate that learners should independently engage regulatory activities such as planning, monitoring and evaluation, research has consistently shown that additional support is needed for learners to become ''active agents of their own inquiry learning'' (Kluwe 1982 quoted by van Joolingen et al. 2005). Thus, it might be useful to limit the users' access because of the variety of options. To prevent students from unintentionally deleting objects and already achieved model states it can be chosen to restrict or take away the edit access from the simulator. For instance, the user can run simulations and store result but cannot carry out editing actions (Byrknes and Myrtveit 1996). This kind of exploratory learning, which is learning with given models, can also be offered by the presented ELS.

[2] The basic structure of a formal SD-model is a system of coupled, nonlinear, first-order differential (or integral) equations.

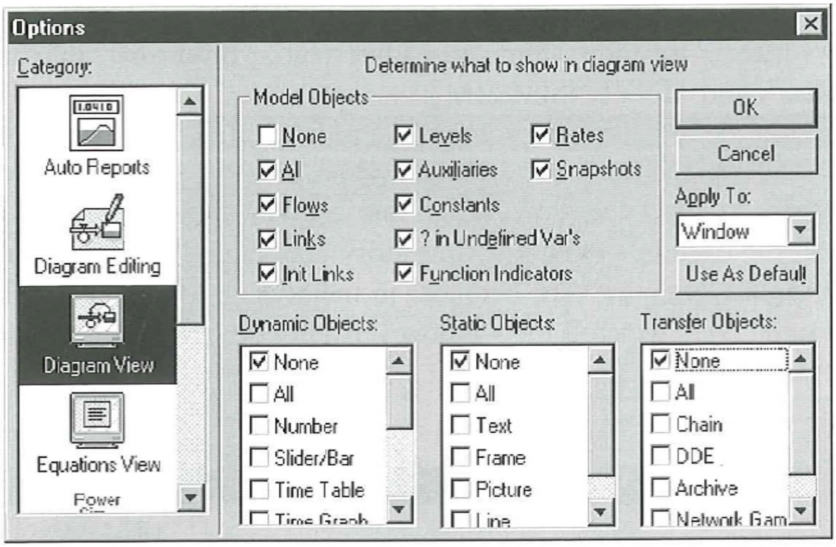


Fig. 16.4 Option window for the simulation results as visualization features (screen shot)

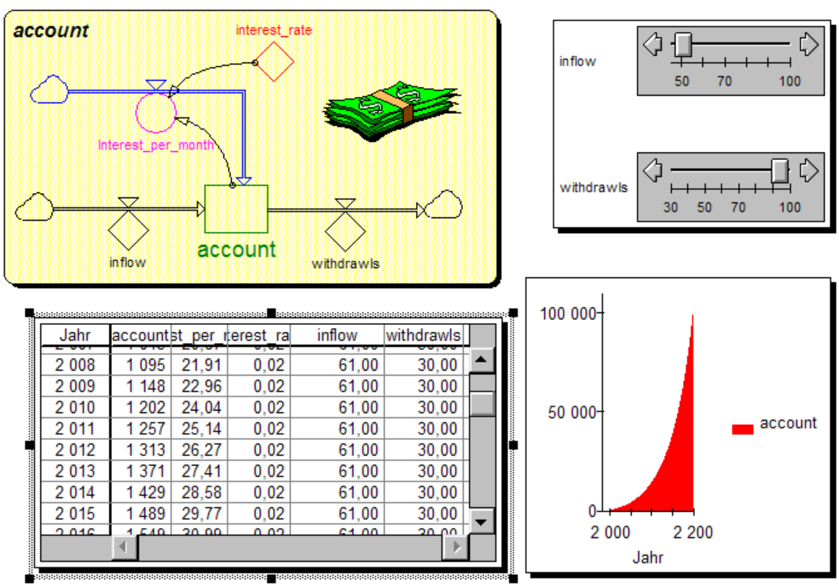


Fig. 16.5 Output and control window for the simulation results (screen shot)

16.4 Methods and Techniques of Inquiry

16.4.1 Diagnostics and Analysis

Concept map approaches are targeting on the representation of knowledge which is acquired by a learner. Specifically, the approaches focused on what students have learned, how much, how structured or isolated the knowledge is. One of the research questions in this study is to determine the level of processing (i.e. higher order thinking) students can acquire to their knowledge using the SD-based ELS.

Data have been collected from 89 students over a time period of two years in VET- schools. The students were asked to express their knowledge about the subject matter verbally on worksheets as well as using the SD-based models. The subject matter focused on 5 different but connected learning units and these are marketing, inventory, personnel/human resources, production and finance. To assess the learning outcomes with their different representational format a system of categories of knowledge has been developed (see Table 16.1).

The defined and operationalized categories of knowledge (CK) refer to the definition of policy concepts in accordance to the SD-methodology (Sterman 2000; Forrester 1972). There are two main distinctive categories: knowledge about the system, (i.e. on concepts, structural knowledge, and its interrelations) and complex concepts referring to knowledge about policies.

Table 16.1 System of categories of knowledge (CK1-14)

Type of Knowledge	Category	Label of the Category
Knowledge about the System CK1-8	Category 1	Concepts
	Category 2	Simple linear structured knowledge
	Category 3	Chained linear structures knowledge
	Category 4	Chained l. structured knowledge with exogenous time delays
	Category 5	Knotted structured knowledge
	Category 6	Knotted structured knowledge with exogenous time delays
	Category 7	Circular structured knowledge
	Category 8	Circular structured knowledge with exogenous time delays
Policy Concepts (Higher order knowledge CK9-14	Category 9	Action knowledge
	Category 10	Action knowledge with exogenous time delays
	Category 11	Simple policy concepts
	Category 12	Simple policy concepts with exogenous time delays
	Category 13	Constraint complex policy concepts
	Category 14	complex policy concepts

The learning results were compiled via a verbal protocol analysis approach (Früh 2001). The reliability of this approach was analyzed with the inter-rater-reliability coefficient (CR) of Früh. (2001). The variable *Ü* is equal the number of congruent codes whereas C1/C2 is equal the total number of codes coded by the rater 1 and 2 (see equation 16.1). The range of the CR may vary between 0.0 and 1.0. The CR will be high when there is little variation between the scores given to each item by the raters. In this study the CR indicated a value of 0.787. The value of 0.787 proved that different researchers applying the tool for the analysis (see Table 16.1) have received almost congruent results in coding. In this case the reliability coefficient expresses the agreement among the raters concerning the individual CK the students acquired. Additional statistical analysis and tests were conducted (i.e. the formal and actual pre-knowledge) to enhance and confirm the validity of the results (see Sect. 16.5.2, Hillen 2006a). Other co-variables were included like teachers' competence (Hillen 2004). To present the whole research design would have exceeded the scope of the paper; it represents a rather didactic-methodical research design (Hillen 2006b)[3].

$$CR = \frac{2\ddot{U}}{C_1 + C_2} \tag{16.1}$$

16.4.2 Data Sources and Materials

Data sources used for the analysis were the saved dynamic concept maps (SD-models) as well as filled out worksheets. The worksheets were used as a guide to knowledge construction activities while students were actively constructing dynamic concept maps. Hence, these different data sources represent the learning-process results in two different knowledge representation formats: a verbal format (what the students have written on the worksheets) and a graphical format (the dynamic concept maps). Both data sources are related to the same learning units.

16.4.2.1 Worksheets

The worksheets are seen as external (pre)requisites (tools) of the ELS. They guide and initialize the learning process. The initial and decisive aspect is the case given to evoke cognitive dissonance as well as providing orientation to existing concepts and terms in the related subject matter proposed by the accounted curriculum.

[3] Hillen, S. (2006b). Zum Erwerb generischer Erklärungsmuster zu kaufmännischen Sachverhalten in Orientierung an ein systemdynamisches Modellunternehmen. In: *bwp@Berufs- und Wirtschaftspädagogik – online,* 11. Online http://www.bwpat.de/ausgabe10/hillen_bwpat10.shtml. Accessed 20 August 2011.

The worksheets are designed and used accordingly to the concept of the 'elaboration of mental models' to enable students to develop and apply higher order thinking procedures. This didactical structure is incorporated in all worksheets guiding learning within the SD-based ELS.

Related to the problem discussed in Sect. 16.3.3, the overwhelming task for some students to self-regulate their learning the worksheet use the idea of *complexity progression*. This idea is adapted from the work of de Jong and van Joolingen 1998. This so-called model progression allows learners to start with the construction of a simplified version of the simulation. The complexity is gradually increased by introducing new variables or features. In contrast to de Jong and van Joolingen's approach the increasing complexity in this study is not controlled by new variables but by changed events taking place through the given case (e.g. new competitors, seasonal shifts in demand etc.). Thus, the worksheets have more than just one function: they support teaching and learning, provide a didactical structure to allow self-regulated and reflective learning, as well as the worksheet provide data for assessment. The discussion about the worksheets construction process does not continue here because it is genuine didactical one (Hillen 2011)[4].

16.4.2.2 SD-Models and Their Analysis

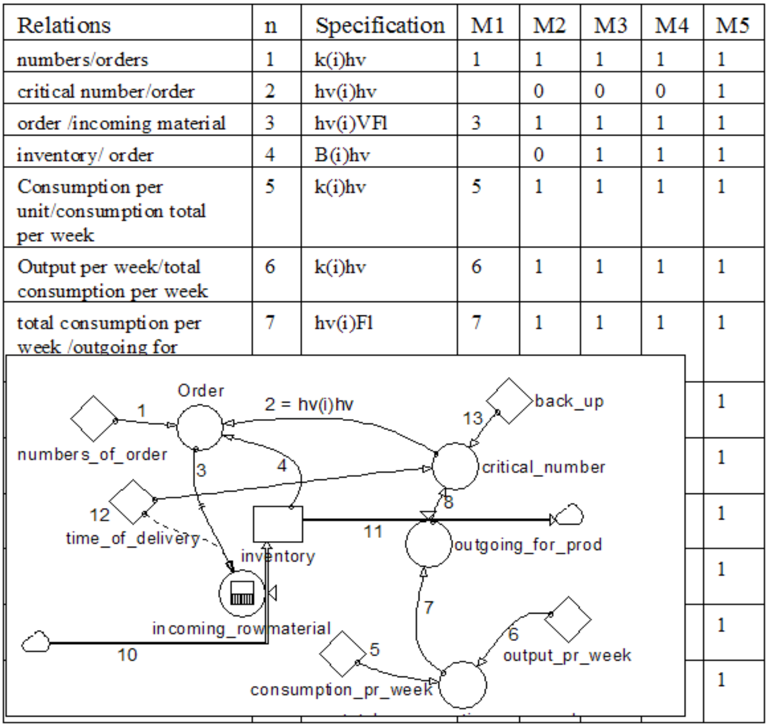
A SD-model is a graphical description of a student related to a certain stage of a learning task. The coding process uses the CK listed in Table 16.1, for analyzing the dynamic model. The coding process takes every variation, every step or change into consideration during the construction process with the exception of repetitions. This is explained by the mental model approach: A mental model is viable until new 'information' leads to a cognitive conflict and the mental model will be (re)constructed or adjusted accordingly. By tracing the models stages it could be recognized that a simulation lead to a reconstruction of the models. The simulation results showed the students the viability of their models. This confirms the approach of the elaboration of mental models by modeling activities.

A prior research approach showed that simulation leads to a 'shift' of elaboration of student's knowledge (Hillen 2004). For this a reference model was used to identify the shifts of elaboration after a simulation approach as well as the goodness of the model's correspondence (see Fig. 16.6). Two issues emerged: The first one concerns the business world is individual, neither enterprise behavior nor any organizational structure will look alike. Even if several reference models would have been developed some appropriate models would have become rejected.

The second issue was the quality of the SD-model which was then reduced by a structural analysis (Hillen 2004). The aspect of dynamic, such as the feedback loops would have been lost. Therefore this approach was not further elaborated.

---

[4] Hillen, S. (2011). The role of worksheets in media based instruction - a didactic and diagnostic approach. In S. Hillen, T. Sturm, I. Willbergh (Eds.) *Challenges facing contemporary didactics*. (pp. 169-184). Münster: Waxmann.



Relations	n	Specification	M1	M2	M3	M4	M5
numbers/orders	1	k(i)hv	1	1	1	1	1
critical number/order	2	hv(i)hv		0	0	0	1
order /incoming material	3	hv(i)VFl	3	1	1	1	1
inventory/ order	4	B(i)hv		0	1	1	1
Consumption per unit/consumption total per week	5	k(i)hv	5	1	1	1	1
Output per week/total consumption per week	6	k(i)hv	6	1	1	1	1
total consumption per week /outgoing for	7	hv(i)Fl	7	1	1	1	1
							1
							1
							1
							1
							1
							1

Fig 16.6 The primarily structural reference system (screenshot)

The diagnostic study set by Verburgh 1996 was not applied because the analysis did not include content and structure at once. Such an approach used several steps and methods to analyze the students' results. It holds a pre and post tests; whereas, my study explicates how the learning is a sum of learning processes. In addition, a diagnostic tool (see Fig.16.6) based on the principle of content analysis respecting content and structure was built. With this tool the dynamic concept maps and the written verbalizations on the worksheets were coded and analyzed.

16.5 Results of the Application of SD-Based ELS

16.5.1 Learning Outcome and Representation Formats

Knowledge representation and assessment were used to set ex-post 'knowledge maps' out of learning results, from verbal protocols, to reveal the degree of structural knowledge or the complexity of the knowledge acquired by learners.

In this approach, students constructed dynamic models by themselves during lessons at VET-schools mirroring the (re)constructed knowledge. The graphical learning explicates the SD-models reveal the knowledge acquired as the sum of the just-in-time constructed artifacts (see Fig. 16.7). Likewise students had access to them because they are saved by themselves. Hereby students could evaluate the appropriateness of these models over time because they are simulatable. Finally, learners were able to compare their results simply with their textbooks.

Alternatively, students were using the related online business simulation platform referring to the curriculum. The simulation and assessment are documented in Molkenthin et al. 2008. The business simulation used at the end of the schooling year was used to ensure the application of the acquired knowledge as well as to increase the complexity in learning. A learning unit focused on just one activity of an enterprise; whereas, the business simulation offers a holistic approach. The topics of the learning units were congruent to the activities in the enterprise simulation. The results show the change from naïve mental models (everyday knowledge) to knowledge about appropriate business activities (Berendes 2002).

The qualitative-learning results were studied through a verbal protocol analysis (Früh 2001) by a coding process respecting system dynamics criteria for higher order thinking differentiated in categories of systems knowledge and policies (see Table 16.1). Fig. 16.7 represents the learning outcomes of the individually incremental constructed SD-models compared to verbal-learning explicates (taken from the worksheets). Written tests were not necessary because the worksheets function was guidance and assessment at once. Moreover, the learning results represent already applied knowledge the students acquired while constructing and simulation on business administration subject matters by solving cases.

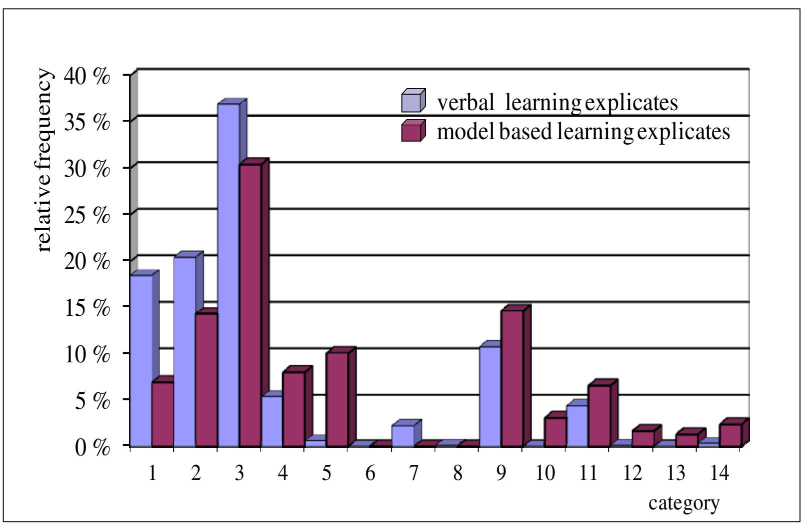


Fig 16.7 Total of learning explicates – verbal and graphical representation formats

These results of the verbal protocol analysis sensu Früh were statistically tested with a non-parametric sign test (see Table 16.2). It shows a significant statistical difference between both representational formats of the models and the verbalizations. A higher coding for the application of graphical representation ($3b = 29$) can be observed. The p-value is <0.001. That means that the application of SD-modeling, in an integrated ELS, supports the acquisition of higher order knowledge. It has to be stressed that just appropriate constructed knowledge to business subject matters were analyzed; that is, flawed representations were not coded.

16.5.2 Learning Outcome and Pre-Knowledge

Beside the analyses of the learning outcomes (explicates) of different representation formats, an analysis of variance related to the acquisition of higher order knowledge was conducted. The independent variables, which have been analyzed, are the teaching method (Method), the competence of the teachers' (Lehrerkp), the formal pre-knowledge (Formalab), and the interplay between those variables (see Table 16.3). The selected hypotheses focusing on the paper's topic are:

- Ha1: The quality of the learning outcomes varies with the level of the teacher's competence.
- Ha2: The quality of the learning outcome varies with the pre-knowledge of the students.

The reason for choosing pre-knowledge as independent variable as well as the teachers' competence is based in research results showing that pre-knowledge influences learning outcomes (Dochy 1999). Moreover, students with lower pre-knowledge are using less effective self-regulating strategies applying ELS than those students with higher pre-knowledge (Winters et al. 2008). The analysis of variance (ANOVA) shows that the level of the teacher's competence (expertise) leads to significant different learning results of higher order knowledge.

Table 16.2 A non-parametric sign test

Statistics for non-parametric sign test on model and verbal learning explicates	
1) Z	-3,5
2) Asymptotic significance (2-sides)	,000
frequencies (model-verbal)	
3a) Negative differences (model<verbal)	7
3b) Positive differences (model>verbal)	29
3c) Bindings (model = verbal)	53
4) Total	89

Table 16.3 ANOVA - dependent variable: learning explicates (CK9-14)

Source	Sum squares type III	df	Average of squares	F	Signif.
1) Corrected model	69,393	11	6,308	3,710	,001
2) Constant term	468,207	1	468,207	275,385	,000
3) Method	16,864	1	16,864	9,919	,003
4) Lehrerkp	23,617	2	11,808	6,945	,003
5) Formalab	14,189	3	4,739	2,782	,54
6) Method*Lehrerkp	,000	0	-	-	-
7) Methode*Formalab	,862	2	,431	,253	,777
8) Lehrerkp*Formalab	1,956	3	,652	,383	,765
9) Meth*Lehrerkp* Formalab	,0000	0	-	-	-
10) Error	64,607	38	1,700	-	-
11) Total	646,000	50	-	-	-
12) Corrected Total Variation	134,000	49	-	-	-

The *p*-value of 0,003 is to be halved because the hypothesis is to be treated as a directed hypothesis. Thus, the *p*-value is 0.0015. The alternative hypothesis *Ha1* is accepted. The pre-knowledge of the students leads to significant different learning results of higher order knowledge. The *p*-value of 0.054 has to be halved because the hypothesis is to be treated as a directed hypothesis. Thus, the p-value is 0.027. The alternative hypothesis *Ha2* is accepted. It has to be stressed that for this Anova just the learning outcomes of higher order thinking (CK 9-14) have been used. In the beginning of the study a pre-knowledge test called 'Test of Economic Literacy` (TEL) were conducted (Beck and Krumm 1998). These test results applying the TEL did not confirm the dependencies of learning outcomes on the student's pre-knowledge but this analysis included all categories of students' acquired knowledge (CK 1-14).

16.6 Conclusions and Future Trends

This investigation tried to show that the SD-based integrated ELS contributes to the acquisition of higher order knowledge by offering and supporting SRL opportunities through the elaboration of mental models (e.g., the features of simulation and feedback). If revisiting the quotation of Zimmerman 1989: "Students can be described as self-regulated to the degree that they are metacognitively, motivationally, and behaviorally active participants in their own learning process", it can be concluded: To support metacognitive learning processes the individual learner needs a platform to apply his strategies (e.g., reflecting his learning achievement).

These processes require a technology and a didactically based framework to reveal the incongruence of the naïve mental model which leads to cognitive conflicts and which motivates to proceed further. The SD-model mirrors the actual understanding of the subject matter because it's actively constructed by the students themselves. These processes wouldn't be possible to support without the visualizations features (table, graph, etc.) of the ELS beside others presented material (e.g., worksheets guarantying enough cognitive conflicts as well as offering a didactical structure) and physical elements of the ELS. The following paragraph will discuss firstly, the results of applying different representation formats to express higher order knowledge, secondly the results of the ANOVA using learning outcome as dependent variable.

One conclusion of the first part of the study is: The different representation formats lead to different opportunities to express what students have learned. Fig. 16.7 shows the verbal explicates derived from the worksheets (blue columns) indicates predominantly declarative concepts in the main category of 'knowledge about the system' in categories CK 1-8. The learning explicates of the SD-models (red columns) additionally hold higher order knowledge concepts, 'policies' represented in categories CK 9-14. Referring to discussion in the beginning of the paper (Oerter 1971; Dörner 1976) it seems that a language assumed to offer a notation supporting to express complex knowledge actually contributes to express more differentiated knowledge concepts. That is, as a result of applying the graphical notation leads to express significantly higher order knowledge than expressing it verbally. Hence, an ELS should consider the choice of language offered students for learning. However, it is not decisive which kind of knowledge representation format might be the 'better' one to support learning, but which kind of representation format is more appropriate to support a specific category of knowledge. Hence, it could be useful in a discipline to acquire knowledge about the system and in other subject matters to develop an understanding of policy concepts (higher order knowledge).

The second part of the discussion focuses on the self-regulation skills and the interplay with pre-knowledge and the acquisition of knowledge. Several studies have been investigating pre-knowledge related to the use of self-regulated strategies. The assumption is that pre-knowledge plays a critical role in the forethought or planning phase of SRL (Pintrich 2000) (see Sect. 2.4). Other researchers (Moos and Azevedo 2008) found out that students used significantly more planning and monitoring with higher domain specific pre- knowledge than students with lower pre-knowledge. This revealed also that students with lower pre-knowledge used more passive strategies. The study of Winters et al. 2008 found in particular the evidence that students who are more academically successful tend to use more effective strategies when learning with an ELS than those with lower pre-knowledge. In addition, their investigations showed that students with higher pre-knowledge tend to engage in greater instances of planning and monitoring.

With the aim to summarize the results of experts (Moos and Azevedo 2008) in this field it can be said that: Students with lower pre-knowledge are using only a few, specific strategies such as summarizing and note taking.

Those students seldom apply active strategies such as making inferences or elaborating on their knowledge. Students with higher pre-knowledge tended to have an internal locus of control which indicates higher self-regulation (MacGregor 1999). Trying to combine these findings with this investigation one may come up with the following hypotheses:

- Learners with higher pre-knowledge have a stronger bias to apply self-regulation strategies which have a rather active character (monitoring, elaborating and inferencing) are likely to achieve higher order knowledge,
- Learners with lower pre-knowledge have a stronger bias to apply self-regulation strategies which have a less active character (summarizing or highlighting) are likely to achieve less high order knowledge.

Such derived hypotheses might appeal to conduct further research. A future trend can be seen that beside improved technology and intelligent features in ELS to use specific intelligent features of an ILS to detect the learners needs. Research on these issues has started in technical and dynamic domains like air traffic control whereas metacognition is simulated as a component (Josyula et. al 2010).

These kinds of learning environment make use of the combination of elements containing external (technology and tools) and internal prerequisites of learning. Internal prerequisites like pre-knowledge, self-regulating learning skills and expertise are provided by students or users as well as by teachers and modelers themselves. The necessities are to investigate further this interplay to be able to satisfy the different needs of individuals involved.

Acknowledgments. The research project was supported by the German Research Association (DFG B-1045/1-5).

References

Alessi, S.: Modeling a system and teaching a system: Enhancing what students learn from modeling. In: Blumschein, P., Hung, W., Jonassen, D., Strobel, J. (eds.) Model-based Approaches to Learning: Using Systems Models and Simulations to Improve Understanding and Problem Solving in Complex Domains, pp. 199–214. Sense Publishers, Rotterdam (2009)

Ausubel, D.P.: The use of advance organizers in the learning and retention of meaningful verbal material. Journal of Educational Psychology 51, 267–272 (1960)

Bandura, A.: Self-efficacy: The exercise of control. W.H. Freeman, New York (1997)

Beck, K., Krumm, V.: Wirtschaftskundlicher Bildungs-Test (WBT). Handanweisung. Hogrefe, Göttingen (1998)

Berendes, K.: Lenkungskompetenz in komplexen ökonomischen Systemen. Wiesbaden, Gabler (2002)

Beschluss der, K.M.K: Rahmenlehrplan für den Ausbildungsberuf Industriekaumann/Industriekauffrau, vom. 9 (Juni 1995)

Beschluss der, K.M.K.: Rahmenlehrplan für den Ausbildungsberuf Industriekaufmann/Industriekauffrau, vom. 14 (Juni 2002)

Bliss, J.: From mental models to modeling. In: Mellar, H., Bliss, J., Boohan, R., Ogborn, J., Tompsett, C. (eds.) Learning with Artificial Worlds, pp. 27–32. The Falmer Press, London (1994)

Byrknes, A.H., Myrtveit, M.: Learning dynamic modeling. Powersim Press (1996)

Collins, A.M., Quillian, M.R.: Retrieval time from semantic memory. Journal of Verbal Learning and Verbal Memory 8, 240–247 (1969)

Dann, H.D.: Variationen von Lege-Strukturen zur Wissensrepräsentation. In: Scheele, B. (ed.) Struktur-Lege-Verfahren als Dialog-Konsens-Methodik, pp. 3–38. Aschendorf, Münster (1992)

de Jong, T.: Instruction based on computer simulations. In: Mayer, R.E., Alexander, P.A. (eds.) Handbook of Research on Learning and Instruction, pp. 446–466. Routledge Press, New York (2010)

de Jong, T., van Joolingen, W.R.: Scientific discovery learning with computer simulations of conceptual domains. Review of Educational Research 68, 179–201 (1998)

Dochy, F., Segers, M., Buehl, M.: The relation between assessment practices and outcomes of studies: The case of research on prior knowledge. Review of Educational Research 69, 145–186 (1999)

Dörner, D.: Problemlösen als informationsverarbeitung. Kohlhammer, Stuttgart (1976)

Forrester, J.W.: Grundzüge einer systemtheorie (Principles of systems). Gabler, Wiesbaden (1972)

Forrester, J.W.: Industrial dynamics. Productivity Press, Portland (1961)

Früh, W.: Inhaltsanalyse, theorie und praxis. 5. UVK Verlag, Aufl. Konstanz (2001)

Hillen, S.: The role of worksheets in media based instruction: A didactic and diagnostic approach. In: Hillen, S., Sturm, T., Willbergh, I. (eds.) Challenges Facing Contemporary Didactics, pp. 169–184. Waxmann, Münster (2011)

Hillen, S., Breuer, K., Tennyson, R.: Gaming and learning: Theory, research, and practice. In: Hillen, S., Sturm, T., Willbergh, I. (eds.) Challenges Facing Contemporary Didactics, pp. 1185–1198. Waxmann, Münster (2011)

Hillen, S.: Die Abbildung von qualitäten des wissens zu kaufmännischen sachverhalten ein inhaltsanalytischer Zugang. In: Minnameier, G., Wuttke, E. (eds.) Berufs und wirtschaftspädagogische Grundlagenforschung. Lehr-Lern-Prozesse und Kompetenzdiagnostik, pp. 377–389. Lang, Festschrift für K. Beck. Frankfurt a. M. (2006a)

Hillen, S.: Zum erwerb generischer erklärungsmuster zu kaufmännischen sachverhalten in orientierung an ein systemdynamisches modellunternehmen. In: bwp@ Berufs und Wirtschaftspädagogik (2006b), `http://www.bwpat.de/ausgabe10/hillen_bwpat10.shtml` (accessed August 20, 2011)

Hillen, S.: Systemdynamische modellbildung und simulation im kaufmännischen unterricht: Elizitation und elaboration von mentalen modellen in komplexen betriebswirtschaftlichen gegenstandsbereichen. Dissertation. Konzepte des Lehrens und Lernens. 10. Peter Lang Verlag, Frankfurt/Main (2004)

Hillen, S., Berendes, K., Breuer, K.: Systemdynamische modellbildung als werkzeug zur visualisierung, Modellierung und diagnose von wissensstrukturen. In: Mandl, H., Fischer, F. (eds.) Wissen sichtbar machen: Begriffsnetze als werkzeuge für das wissensmanagement in lehr- und lernprozessen, pp. 71–89. Hogrefe, Göttingen (2000)

Johnson-Laird, P.N.: The computer and the mind. An introduction to cognitive science. University Press, Cambridge (1988)

Johnson-Laird, P.N.: Mental models: Towards a cognitive science of language, inference and consciousness. University Press, Cambridge (1983)

Jonassen, D.H.: What are cognitive Tools? In: Kommers, P.A.M., Jonassen, D.H., Mayes, J.T. (eds.) Cognitive Tools for Learning, pp. 1–6. Springer, Heidelberg (1991)

Josyula, D.P., Hughes, F.C., Vadali, H., Donahue, B.J., Molla, F., Snowden, M., Miles, J., Kamara, A., Maduka, C.: Metacognition for self-regulated learning in a dynamic environment. In: Proceedings of IEEE SASOW, pp. 261–268 (2010)

Kluwe, R.H.: Cognitive knowledge and executive control: Metacognition. In: Griffin, D. (ed.) Animal Mind – Human Mind. Springer, New York (1982)

Kluwe, R.H., Haider, H.: Modelle zur internen repräsentation komplexer technischer systeme. Sprache und Kognition 9(4), 173–192 (1990)

Kolloffel, B., Eysink, T.H.S., de Jong, T.: The role of external representations in learning combinatorics and probability theory. In: Verschaffel, L., de Corte, E., de Jong, T., Elen, J. (eds.) Use of External Representations in Reasoning and Problem Solving, pp. 169–191. Routledge Press, Abingdon, Ox (2010)

Lajoie, S.P., Azevedo, R.: Teaching and learning in technology-rich environments. In: Alexander, P.A., Winne, P.H. (eds.) Handbook of Educational Psychology, 2nd edn., pp. 803–821. Erlbaum, Mahwah (2006)

Mandl, H., Spada, H.: Wissenspsychologie: Einführung. In: Mandl, H., Spada, H. (eds.) Wissenspsychologie, pp. 1–16. Psychologie Verlags Union, Münche (1988)

MacGregor, S.K.: Hypermedia navigation profiles: Cognitive characteristics and information processing strategies. Journal of Educational Computing Research 20(2), 189–206 (1999)

Molkenthin, R., Breuer, K., Tennyson, R.D.: Real-time diagnostics of problem solving behavior for business simulations. In: Baker, E.L., Dickieson, J., Wulfeck, W., O'Neil, H.F. (eds.) Assessment of Problem Solving Using Simulations, pp. 201–221. Lawrence Erlbaum Associates, London (2008)

Moos, D.C., Azevedo, R.: Self-regulated learning with hypermedia: The role of prior domain knowledge. Contemporary Educational Psychology 33(2), 270–298 (2008)

Norman, D.A.: Cognitive engineering. In: Norman, D.A., Draper, S.W. (eds.) User-Centered Design, pp. 31–62. Erlbaum, Hillsdale (1986)

Norman, D.A.: Some observations on mental models. In: Gentner, D., Stevens, A.L. (eds.) Mental Models, pp. 7–14. Erlbaum, Hillsdale (1983)

Oerter, R.: Psychologie des denkens. Auer, Donauwörth (1971)

Pintrich, P.R.: The role of goal orientation in self-regulated learning. In: Boekaerts, M., Pintrich, P.R., Zeidner, M. (eds.) Handbook of Self-Regulation, pp. 451–502. Academic, San Diego (2000)

PZ-Informationen. Nichtlineare dynamische Systeme und Chaos. Handreichungen zum neuen Lehrplan Physik in der SII, 3. PZ, Bad Kreuznach (2000)

Ryssel, J., Sommer, S., Fürstenau, B., Kunath, J.: The effect of different concept mapping techniques on promoting students' learning processes in the field of business. In: Canas, A.J., Reiska, P., Ahlberg, M., Novak, J.D. (eds.) Proceedings of CMC Concept Mapping: Connecting Educators, pp. 238–241 (2008)

Salomon, G.: AI in reverse: Computer tools that turn cognitive. Journal of Educational Computing Research 4(2), 123–139 (1988)

Salomon, G., Perkins, D.N., Globerson, T.: Partners in cognition: Extending human intelligence with intelligent technologies. Educational Researcher 4, 2–9 (1991)

Scheele, B., Groeben, N.: Dialog-konsens-methoden zur rekonstruktion subjektiver theorien (Structure-laying technique for reconstructing subjective theories "every day-life"-theories). Francke, Tübingen (1988)

Schunk, D.H., Zimmerman, B.J. (eds.): Self-regulated learning: from teaching to self-reflective practice. Guilford Press, New York (1998)

Simons, P.R.J.: Lernen selbständig zu lernen - ein rahmenmodell. In: Mandl, H., Friedrich, H.F. (eds.) Lern- und denkstrategien: Analyse und intervention, pp. 251–264. Hogrefe, Göttingen (1992)

Spector, J.M., Christensen, D.L., Sioutine, A.V., McCormack, D.: Models and simulations for learning in complex domains: Using causal loop diagrams for assessment and evaluation. Computers in Human Behavior 17, 517–545 (2001)

Sterman, J.D.: Business dynamics: Systems thinking and modeling for a complex world. McGraw-Hill, Boston (2000)

Sweller, J.: Cognitive load during problem solving: Effects on learning. Cognitive Science 12, 257–285 (1988)

Ulrich, H., Probst, G.J.B.: Anleitung zum ganzheitlichen denken und handeln 4. Haupt, Aufl. Wien (1995)

van Borkulo, S.P., van Joolingen, W.R., Savlesbergh, E.R., de Jong, T.: A framework for the assesssment of learning by modeling. In: Blumschein, P., Hung, W., Jonassen, D., Strobel, J. (eds.) Model-based approaches to learning: Using systems models and simulations to improve understanding and problem solving in complex domains, pp. 179–195. Sense Publishers, Rotterdam (2009)

van Joolingen, W.R., Lazonder, A.W., de Jong, T., Savlesbergh, E.R., Manlove, S.: Co-Lab: Research and development of an online learning environment for collaborative scientific discovery learning. Computers in Human Behavior 21, 671–688 (2005)

Vandevelde, S., van Keer, H., de Wever, B.: Exploring the impact of student tutoring on at-risk fifth and sixth graders' on self-regulated learning. Learning and Individual Differences 21, 419–425 (2011)

Verburgh, L.D.: Participative policy modeling applied to the health care insurance industry. Benda, Nijmegen (1996)

Willerman, M., MacHarg, R.A.: The concept map as an advance organizer. Journal of Research in Science Teaching 28, 705–711 (1991)

Winters, F.I., Greene, J.A., Costich, C.M.: Self-regulation of learning within computer-based learning environments: A critical analysis. Educational Psychology Review 20(4), 429–444 (2008)

Zimmerman, B.J.: Investigating self-regulation and motivation: Historical background, methodological developments and future prospects. American Educational Research Journal 45, 166–183 (2008)

Zimmerman, B.J.: Attaining self-regulation: A social cognitive perspective. In: Boekaerts, M., Pintrich, P.R., Zeidner, M. (eds.) Handbook of self-regulation, pp. 13–39. Academic, San Diego (2000)

Zimmerman, B.J.: Models of self-regulated learning and academic achievement. In: Zimmerman, B.J., Schunk, D.H. (eds.) Self-regulated Learning and Academic Achievement. Theory, Research and Practice, pp. 1–25. Springer, Heidelberg (1989)

Zimmerman, B.J., Schunk, D.H.: Self-regulated learning and academic achievement. Theoretical perspectives, 2nd edn. Laurence Erlbaum Associates, Mahwah (2001)

Abbreviations

ANOVA	Analysis of Variance
CK	Categories of Knowledge
CR	Coefficient of Interrater-Reliability
ELS	Educational Learning System
SRL	Self-Regulated Learning
SD	System Dynamics
TEL	Test on Economic Literacy

Chapter 17
Seamless Web-Mediated Training Courseware Design Model: Innovating Adaptive Educational-Learning Systems

Elspeth McKay[1] and John Izard[2]

[1] School of Business Information Technology and Logistics, RMIT University
GPO Box 2476V, Melbourne, Victoria 3106 Australia
elspeth.mckay@rmit.edu.au
[2] School of Education, RMIT University
GPO Box 2476V, Melbourne, Victoria 3106 Australia
john.izard@rmit.edu.au

Abstract. We present an innovative Web-mediated training system design architecture that encourages novice courseware developers to deliver their own adaptive (user-centered) *educational-learning systems* (AELS) by utilizing achievable and cost effective online training modules. We propose a seamless Web-mediated training system design architecture that includes a choice of intelligent agents (personal Avatars) to guide the trainee through their knowledge acquisition. We cut through the more usual *information systems* (IS) development rhetoric. Instead of cloaking the courseware design process in highly technical mystery, we argue that educational technology experts should encourage non-technical developers to believe that the possibility of customizing their own Web-mediated training programs falls within their grasp. The preliminary findings from a pilot study conducted to test our AELS as an in-house courseware development tool indicate that it will be most suitable for corporate and government training courseware creators.

17.1 Introduction

While this chapter explores the process component of operational design (van Bennekom 1995), its main purpose is to outline a systematic approach to user-centered *educational-learning systems* (ELS) design that is easy to reproduce as an adaptive Web-mediated training design model, one that interacts with the trainee. By user-centered, the learner's preference for learning context is set. Along the way we will show how some of the usual technical barriers may not be difficult for novice Web-content developers to overcome when using this approach.

We believe that simplifying the process may improve the adoption rate of instructional *information communication technology* (ICT) tools, which has not kept pace with technological advances.

A. Peña-Ayala (Ed.): Intelligent and Adaptive ELS, SIST 17, pp. 417–441.
springerlink.com

In the rush to reflect society's desire for the latest technology, more appropriate and common sense business solutions are swept up in the resulting chaos (Alonso et al. 2004). Far too often plans for the 'proof of concept' or prototype learning system are drawn up by a non-technical training developer, consequently their specifications ignore the complexities pertaining to the principles of instructional design and the expected functionality to promote user-centered learning is not achieved (McKay & Martin 2009, Merrill 2002).

In our *adaptive educational-learning systems* (AELS) design model we show how to combine Web-hosting service access for corporate/government training courseware creators in a seamless process to provide them with the capability to customize corporate employee training modules, on an 'as needed' basis. Taking such a user-centered approach to learning affords trainees an individual choice as to when they undertake their required training program.

With our *information systems* (IS) design architecture, each learning component can be designed to maximize the interoperability between functional IS requirements. Our approach is unlike other *learning management system* (LMS) architectures, whereby the corporate manager or training facilitator may elect to track a trainee/learner's progress and control what they see and do (Dick et al. 2006). Instead our AELS architecture provides a trainee with the opportunity for complete anonymity to be 'built into' their training sessions.

17.1.1 User-Centered Learning Practice

In the corporate world, online courseware development is shifting away from the traditional notions of adopting ICT tools in face-to-face training sessions, to the design of customizable online workforce training (Schank 2002). As a result, there is an endless list of online course development tools with tantalizingly free training and documentation, for instance: WebEx, Acrobat Connect Professional and Central (Clark and Kwinn 2007). However, unless one has high-end IS skills, mastering these tools requires long lead times.

In recent years the adoption of ICT tools (otherwise referred to as multimedia) has been researched by a diverse mix of professional disciplines. Adoption of ICT tools is defined here as the decision to implement an online solution to solve the need for business workforce training/reskilling. By operational design we mean the set of user-centered activities that people and their computers need to perform upon corporate informational resources.

ICT adoption can be studied at three operational design levels. The first level relates to the need for developers of online training programs to understand how corporate-level investment decisions in online training are made.

The second level applies to the user-centered perspective of the trainee/IS users to explain what people do with their online training/learning experiences. The third level examines the IS design and development process to evaluate why online training adoption rates are not keeping pace with ICT tool advances.

Over the years various terms have been used to describe traditional *computer-based training* (CBT) programs. We define Web-mediated training as any interactive online training/learning program, as described by Driscoll (2001).

Although it may not have been realized at the time, the introduction of ICT tools commenced many years ago. By 1986 more than 30 million people in the USA alone were using computers. Originally people struggled to find the terminology with which to describe this emergent learning technology. *Computer aided learning* (often referred to as CAL) was one of the most commonly used terms to describe computerized learning environments, but nobody was really sure what CAL meant. Instead, people generally recognized that computers could be used to automate learning resources. Confusion continued through the 1990s, when CAL was thought to mean practically anything from a single, monolithic computer program to sets of software applications designed to increase the reach of traditional computerized media. These newly created ICT artifacts included, for example, a lecture series for the educational sector, or industry-based training courses.

Over time educational course developers have added to the confusion as they endeavored to implement more user-centered learning strategies. Table 17.1 represents a matrix, showing that people mix and match the possibilities of using technology in learning programs, in their attempts to launch interactivity between the facilitation media and a participant (Gery 1987). With several was to depict a ICT-enhanced online learning environment, it is difficult to understand what is meant.

IS analysts tend to confuse our collective understanding by interchanging the terms (online learning/distance education/Web-based training) when we really ought to differentiate them with particular user-centered instructional strategies, recognizing that they are very diverse environments. Perhaps one way to understand the complexity of a user-centered CBT environment would be to explain it as, *"An interactive learning experience between a learner and a computer in which the computer provides the majority of the stimulus, the learner must respond, and the computer (then) analyzes the response and provides feedback to the learner"* Gery 1987:7.

Table 17.1 Interactivity of technological media and people

Medium	Action	Environment
Computer	Assisted	Instruction
	Aided	Learning
	Managed	Education
	Based	Training
	Enhanced	Teaching
	Mediated	Development
	Interactive	Study

IS analysts tend to confuse our collective understanding by interchanging the terms (online learning/distance education/Web-based training) when we really ought to differentiate them with particular user-centered instructional strategies, recognizing that they are very diverse environments. Perhaps one way to understand the complexity of a user-centered CBT environment would be to explain it as, *"An interactive learning experience between a learner and a computer in which the computer provides the majority of the stimulus, the learner must respond, and the computer (then) analyzes the response and provides feedback to the learner"* Gery 1987:7.

It is interesting to note that Gery's explanation remains in our vernacular. CBT is difficult to define because we still have not yet embraced the learning potential found in the human-dimensions of effective *human-computer interaction* (HCI) through well placed ICT-pedagogies that promote user-centered instructional strategies. To this end, we face even more choice through the advent of the next generation of interesting, easy to use ICT tools, such as social networking sites. As the area evolves more of these sophisticated tools are added to the mix in a generic Web-developer's techno-grab bag (McKay 2008).

However, not enough people take the time to investigate whether their instructional strategies and delivery media align with the intended training/learning context. When planning to implement an interactive training experience; there is often no regard for the type of learning design, and even less for the instructional architecture. Moreover, there is absolutely no evidence that thought has been given by the courseware creator to the need for user-centered learning practice where people must be given the opportunity to engage in knowledge creation activities.

In the business sector, corporate decisions to invest in an LMS have involved huge amounts of money, in the hope that an LMS will enable online training programs to spill out effortlessly and provide long-term savings. However, for them to be effective, it is essential to differentiate between the type of training required and the technological means to bring forward an effective online instructional architecture to support it. This distinction, which is too often missed, necessitates a closer look at the type of learning activities required in preparing a trainee/learner to develop the desired knowledge and skills.

Perhaps we have become too accustomed to relying upon the constantly improving browser technologies. These sophisticated ICT tools enable platform-independent transmission protocols and ordinary users have no idea of the high-level nature of the interactive technologies that drive their HCI activities online.

We have set the AELS context for this chapter by saying there is confusion surrounding the online training environment. To define our IS design model; we first describe a common view of IS design that is offered to the general public in reference to Web-service design. Web-services currently represent the best vehicle to support our approach to cut down the IS rhetoric which prevents novice courseware developers from achieving successful IS design.

17.1.2 Understanding IS Intelligence

The *behind the scenes* sophistication of ICT tools encourages online interactivity. A range of terminology has evolved to describe an IS that adapts to its users, such as: adaptive interfaces, user modeling systems, software agents and intelligent agents (Jameson 2008). For the uninitiated, these terms may prove to be daunting. In the more traditional ELS sense, an adaptive IS usually means that the learner's ICT usage characteristics are captured by the IS to model or characterize the learner. In order for the IS to be adaptive, the IS perform an automatic *inference* from the information get of the learner's performance and respond accordingly.

Confusion may arise between the meaning of an *adaptable* and an *adaptive* IS (Jameson 2008). An adaptable IS responds to the individual IS user. For instance: when catering for a trainee/learner to choose between a program that provides skill development, which assumes no previous knowledge, and one that is intended as a refresher course. In this case trainees can explicitly customize their training program to their individual preferences. An adaptive IS is one in which the system includes some method for acquiring and exploiting a user model.

To understand how IS intelligence operates within an ELS we combine the fields of cognitive science and *artificial intelligence* (AI). In the former, researchers have discovered ways to depict how people think and solve problems while they learn. AI researchers examine the 'virtual' representation of human knowledge and clarify reasoning and procedural knowledge. Results from cognitive science are utilized by AI experts to create software applications that behave as human beings would. While the application of AI to education has been successful (Forbus and Feltovich 2001), *intelligent tutoring* remains neglected.

Typically *learner modeling systems* combine elements of *user-adaptivity*, CBT and the learning environment (Corbett, et al. 1997) to support learning. Software agents are nothing more than computer programs, albeit sophisticated, in so far as they can act in an 'intelligent' manner. The *intelligence* is written into the software-object to carry out specific tasks. An example of an intelligent agent found on most personal computers would be, for instance, one that monitors computer drives for virus attack.

In creating a courseware creator model our AELS adopts an innovative self-select approach to facilitate a user-centered learning environment (Fig. 17.1). The online courseware is presented in two modes (beginner and experienced courses) and trainees use their regular keyboard arrow keys to select the mode. In the beginner course, training strategies are preset, assuming no knowledge of the task. It carefully guides the trainee through all the learning activity necessary to gain competence. Along the way, learners are given the opportunity to interact with the courseware through a series of interactive learning processes. These activities provide online feedback which guides the instruction according to the learner's input. For the experienced trainee, the courseware template allows the choice of tasks in any order. In each course there is the opportunity for trainees to check their knowledge and skills and a final task requires successful completion.

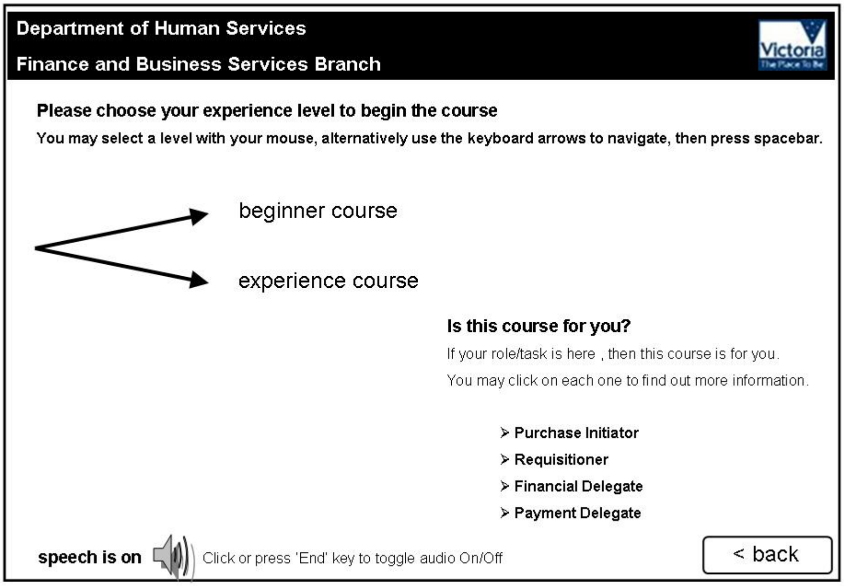


Fig. 17.1 User-centered AELS self-select skill development model

17.2 Seamless Web-mediated Training Systems

The provision of an integrated IS environment is a commendable goal pursued by most corporate IS departments. Despite the good intentions of *sharable content object reference model* (SCORM) compliance, a novice online ELS developer is, more often than not, at a loss to know how to generate these technical standards. A typical SCORM-compliant ELS should consist of individual (learning) content/ICT resource files. Text is quite often kept in separate *eXtensible HyperText Markup Language* (XHTML) files, to be accessed by a standard Web-browser during an online session. HTML, or *hypertext markup language*, has two essential features: hypertext and universality.

Hypertext is the technical term for creating links in Web-pages, which lead visitors to these pages, to other Web-pages, or practically anything else on the Internet. Universality in this context means that the HTML documents are saved as *American Standard Code for Information Interchange* (ASCII) (a high-level machine code) or text-only files that can be read by practically any computer. Adding the *'X'* to the HTML term simply refers to the *extensible markup language* (XML), which can be thought of as a sort of cousin to HTML (Castro, 2000), allowing more flexibility with personalized tagging and the ability to separate content and formatting by using style sheets.

There are many other multimedia file-types that are stored in their respective proprietary formats. Knowing your way around this environment requires specialist IT/IS skills. Even if you know how to use a SCORM editor, using these formats requires an advanced understanding of the SCORM file structure to ensure compliance. This is a technical barrier that prevents most people from producing eLearning content, according to Pohl et al. (2009).

Mindful of the complexities involved when user acceptance of electronic collaboration is problematic (as defined by Dasgupta et al.2002), an IS development team (based in Australia) is working towards reducing technological barriers to enable even novice developers of training systems to produce quality online training/learning programs. Their contribution is to provide a user-centered AELS design model, which is, on the one hand, easily understood by a novice online learning developer; while on the other, provides the necessary high-level IS functionality required to implement a seamless, integrated software development platform. This team consists of two IS academics, a cognitive performance measurement expert, a multimedia developer and a government training courseware developer.

This user-centered Web-mediated AELS architecture involves four interconnected systems (or software layers): *transport, presentation, learning environment and application* (TPLA). Our TPLA model is a viable SCORM-compliant solution which enables the courseware developer to create an adaptive and flexible Web-mediated training environment, particularly for the government/corporate sector. This innovative IS-architecture relies on the integrated configuration of the four TPLA layers that enable a *single-sign-on* (SSO) facility. We describe how our model allows for a user-centered approach to learning in the next section.

17.2.1 Flexibility of Taking a User-Centered Learning Approach

Traditionally, employers view training as an expensive solution that is implemented to fix problems related to knowledge gaps. In the current climate of fast-changing work practices, every time a new ICT tool enters the work environment employers appear to pour endless amounts of money into upgrading their employees' skill base (McKay et al. 2007). This continual investment in workplace learning begs the question of what we know about the impact of these emerging ICT tools on institutional effectiveness. Furthermore, a large number of implemented online training solutions have been poorly designed and inadequately tested. Often paper-based training materials are simply loaded into an LMS or courseware shell (McKay et al 2007); without including adequate knowledge navigation or consideration for the principles of instructional design. Moreover, online learning programs frequently fail to establish whether learning actually occurs (no demonstrated means to check increased proficiency of the participants). In cases where tests are made, most fail to use valid measures of the changes in proficiency (Fahy 2004).

It is no wonder that within corporate and government sectors, online learning solutions are poorly regarded by management and remain unused by employees (McKay et al. 2007); thereby making them ineffective and an expensive waste of limited resources. There are similarities within the education sector, where there is concern for upgrading higher-education graduates' employability skills.

In any instructional/training event it is important to identify the learning domain, and to specify the tasks that are essential for developing the skills and knowledge to achieve the expected learning outcomes (the instructional goals). While the *Dreyfus Model of Skill Acquisition* (DMSA) shows incremental changes in cognitive skill acquisition, in this chapter our use of the same term divides cognitive skill development into two clear sets of cognitive performance. The first set refers to the cognitive skills associated with declarative knowledge (knowing what); the second relates to procedural knowledge (knowing how) (McKay and Merrill 2003). This type of cognitive skill acquisition can be analyzed in four discrete categories (McKay 2008):

1. Verbal information (knowing basic terms),
2. Intellectual skill development (knowing basic rules, discriminating and understanding concepts and principles),
3. Intellectual skill (knowing higher-order-rules, problem solving, the ability to apply concepts and principles in new situations), and
4. Two different types of cognitive strategies: a) to identify sub-tasks, recognize unstated assumptions; b) to recall simple prerequisite rules and concepts, integrating learning from different areas into a plan for solving a problem.

In some instances, these categories of skill can be embedded within the learning content (hierarchical skill development). However, this type of user-centered framework is unsuitable for some other training environments (such as building skills for a specific task). Instead, according to McKay (2007), taking a more user-centered approach to the learning domain should concentrate on the intellectual skills associated with a particular problem-solving issue (such as when a government employee needs to consider whether to invoke the whistleblower legislation, after witnessing unethical behavior in the workplace).

Reasons why people develop new skills or retrain for the workplace are countless. In some cases individuals are merely committed to life-long-learning; while in others a need to develop new skills may be due to unforeseen circumstances: for instance responding to new management needs in an organizational restructure.

It is important to differentiate what individuals know from what they do not. Competency evaluation for both education and corporate training sectors is necessary to define remedial intervention for both new employees and those with more experience, when they are faced with a requirement for vocational retraining.

It is therefore critical for an ELS to provide the courseware creator the added flexibility of an AELS comparable to ours in responding to their choice of training mode (McKay and Martin 2009). Our AELS affords courseware creators a choice of instructional mode that is efficient, reliable and safe to administer, incorporating features of the DMSA for trainees to determine their skill level proficiency.

Adopting this rather new approach to training mode choice benefits from an inbuilt competency assessment that enables the choice of differentiated training modes and provides, for each training task encountered, an adaptive cognitive skills performance measurement that correctly identifies different levels of competency. The benefit of such a competency management system (Anderson 2008) is an increase in opportunities for people to participate in more appropriate online training/retraining programs.

17.3 Designing Educational Learning Systems

To understand the functionality of our AELS, a comparison is made to the IS design layers discussed by Alonso et al. (2004). Essentially they use a linear (drop through) model, saying that the design of an IS involves three linear-layers: presentation, application logic and resource management (Fig. 17.2).

The *presentation layer* performs communication tasks with the client, which may be either human users or other computers. By presenting information to the client/user, the browser interacts with the system by submitting operations and eliciting responses (such as a Web-site homepage or other types of data).

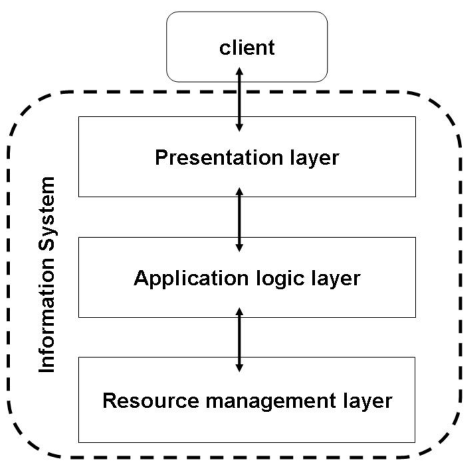


Fig. 17.2 The three layers of an IS (Alonso, et al. 2004:12)

The *application logic layer* processes and delivers information and data (such as the savings withdrawal process of an automated teller machine, or the online enrolment process of a university). According to Alonso et al. (2004), this layer performs the services, sometimes referred to as business processes, business logic, business rules or server.

The *resource management layer* is the information repository that provides the functionality of the database. It deals with and implements the data sources available to the system.

By contrast, we acknowledge the user-centered approach to learning and our AELS extends the Alonso et al. (2004) concept to include an initial *transport layer* that brings, carries and sends the IS protocols to invoke the Web-host. Our *presentation layer* shows the user a graphical interface that presents the training content. The *learning environment layer* provides the interactive Web-mediated training environment. While the *application layer* represents the SCORM-compliant re-purposing/customizing courseware authoring tools used by the Web-content developer.

Our user-centered AELS can be described by either its functional components or the IS-architecture. In our 'functional model' we describe what each layer is designed to do, while our 'AELS architecture' provides a more technical description of these functions. This is a three-tiered IS-architecture, which includes the SSO authentication process, corporate databases and the usually unseen, but necessary, background environments that involve the *Linux-Apache-MySQL-PHP* (LAMP) configuration, generally required for the interoperability of the organization's *content management system* (CMS)-LMS and the Web-training content developer's authoring tool.

17.3.1 Comparison of Interactivity with Existing Systems

According to Peng et al. (2008) the desire of literally millions of people for ubiquitous computing was fulfilled with the advent of increased Internet connectivity. Along with this, researchers sought efficient and effective ways to reach common ground through improving IS standards to orchestrate seamless interactivity online. There are still many difficulties that need to be resolved before the Peng et al (2008) notion of a *ubiquitous-learning system* (ULS) can be achieved. One of the obstructions to this ambitious goal relates to the plethora of computing devices used to access the Internet. For instance: there are the wireless-computing devices that deliver real world environments (personalized services) with combinations of user information and environmental information. To this end Table 17.2 represents the interactive roles that exist between participants and their online IS. Our AELS delivers most of the interactivity described by Peng et al. (2008).

Table 17.2 Interactivity dimensions (adapted from Peng, Chou and Chang, 2008:56)

Role	Interactivity dimension	Definition	AELS
Learners' determination to make a choice in the system	Choice	The amount of the multi-media type of information to which users (learners and instructors) have access	Style of avatar / intelligent agent: textual form, image, video (or none)
	Non-sequential access of choice	Learners can access information in a non-linear way	Choice of content delivery mode (beginner or experienced)
	Monitoring of information use	The system can collect data on the learners themselves, their selections, their use of information and so on. The learners can monitor information regarding them	On the spot feedback to trainees on their progress, with ability for them to redo skill development tasks until successful
System provides services that react to learners' actions	Responsiveness to learners	The system responds to learners' requests in a non-delayed fashion	Immediate real time processing
	Personal choice helper.	Information helps learners make better choices relative to instructional content	User choice with knowledge navigation and avatar profile
	Adaptability	System adapts the interaction process and the exchange of information to individuals	Course selection is left to the trainee – can interact with each course mode if they wish

17.4 Describing the User-Centered AELS Architecture

Our AELS was built implementing the TPLA layers (see Seamless Web-mediated Training Systems). As such, Fig. 17.3 below shows the relationship between the different Alonso et al. (2004) IS design layers and the functional components of our AELS, aligned by arrows between these two conceptual IS models. First we describe the function of each TPLA layer; then we explain the IS architectural model.

17.4.1 The TPLA Layers

17.4.1.1 Transport Layer (T)

The user-centered *transport layer* provides the IS protocols that are necessary to enable the trainee/user to access the organization's Web-site. This software layer allows the trainee/user to retrieve information when the *uniform resource locator* (URL) is entered into their browser (e.g.: http://goal.edu.au). This information is passed through the *hypertext transfer protocol* (HTTP) in the form of a digital

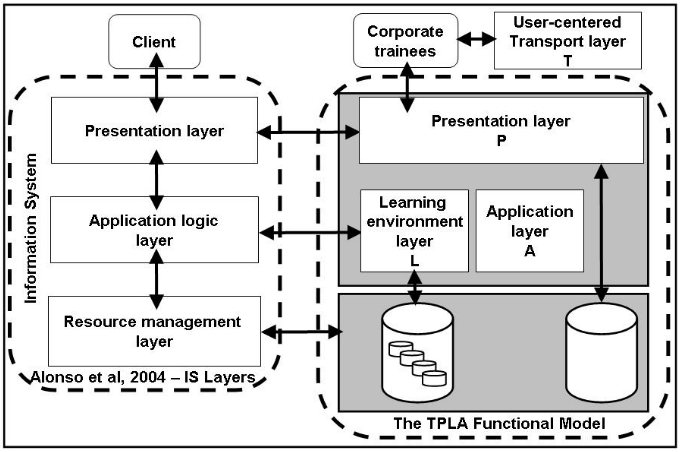


Fig. 17.3 IS design model extended by our TPLA functional model

document that is identified using a *uniform resource identifier* (URI) Alonso et al. (2004). The well-known *World Wide Web Consortium* (W3C) Technical Architecture Group describe the URI as a global IS identifier that links the (IS-level) metadata and data (the screen-based material that people see) to the Web-resource. For the full account of this description see http://www.w3.org/TR/webarch/.

A URI will invoke three Web-based processes: (1) the user identification process that is handled by the URI, which recognizes the IS-address specified in the Web using the URI; (2) the interaction process that are known as the Web-based protocols (e.g.: the URIs, HTTP, IS-messaging and data, communication between the Web and user-agents over an IS-network); and (3) formatting the screen-based data representation retrieval protocols and/or data submission (Jacobs and Walsh, 2004). Generally, such a protocol follows a sequence of one or more IS messages which, when taken together, contain the order of the screen-based material and the IS metadata to transfer information between software agents.

17.4.1.2 Presentation Layer (P)

Entering the training program's URL brings the trainee/user to the agency or corporation's Web-site through the *presentation layer*. This layer provides the IS-communication interface between the Internet (through the T-layer) and external entities (people or other computers). This P-layer displays the screen-based *graphical user interface* (GUI), commonly referred to as the GUI, which is custom-designed to suit the organization. So that the general public cannot gain access to the actual training resources, it is necessary for an authentication process to be installed. This is stored in the organization's database-server and ensures that only authenticated trainers/trainees with IS-rights and privileges gain access to the training program.

17.4.1.3 Learning Environment Layer (L)

The *learning environment layer* has its own user-centered GUI, with an outlook similar to a virtual classroom (Clark and Kwin, 2007). This layer allows three main functions that logically represent the classroom scenario: the IS administrator, training facilitator and trainee. It is possible to include customised courseware design templates for the user interfaces that are specific to the registered user/trainee role. The L-layer allows the user to interact with the training content, conduct quizzes, and populate new content/training materials by using presentation methods such as *'wikis'* and monitor a trainee/user's grades, among other functions. Generally, the IS administrator is tasked to create courseware and make the eResources accessible to trainees/users. The IS administrator role is responsible for high-level IS administration tasks only. Whereas the courseware creator role is responsible for ensuring that courseware will not cause damage to the L-layer. Moreover, as the L-layer defines user roles, we suggest that a user administration role should also be set up to manage the trainee/users into the online courses. Once again – this innovative IS feature reduces the chances of an open slather access approach to the L-layer. The IS requirements for this layer are the Web-server, a scripting language, and a database system similarly installed in the P-layer.

17.4.1.4 Application Layer (A)

The *application layer* represents the SCORM-compliant repurposing system. This system provides an authoring tool (e.g.: ProForm Rapid eLearning Studio, previously called Flashform), that has interoperability with an LMS. This authoring tool operates independently from the other layers, having the capability to display multimedia objects. We claim that training material/content developed with this tool can be customised and saved for reuse.

17.4.2 The AELS Architecture

In keeping with a user-centered approach to learning, the AELS architecture is based on a three-tiered model, shown in Fig. 17.4. This authentication process is made through the SSO facility displayed through the *presentation layer* (a dialogue box requiring a login name and password). By implementing such a SSO, the trainee only needs to sign in once to access the training program. The SSO is necessary for inter-operability functionality between the proprietary CMS (JOOMLA) and the LMS (we use MOODLE). Without this innovative facility, the trainee/user is usually required to undergo the cumbersome process of using two sets of login-password combinations. This may arise because an organization's integrated database-server is often a composition of several databases, managed by a corporate database administrator. In addition, an organization's databases may use different database-servers to store all training profile data, including: activity and interactivity logs, learning objects or modules and assessments.

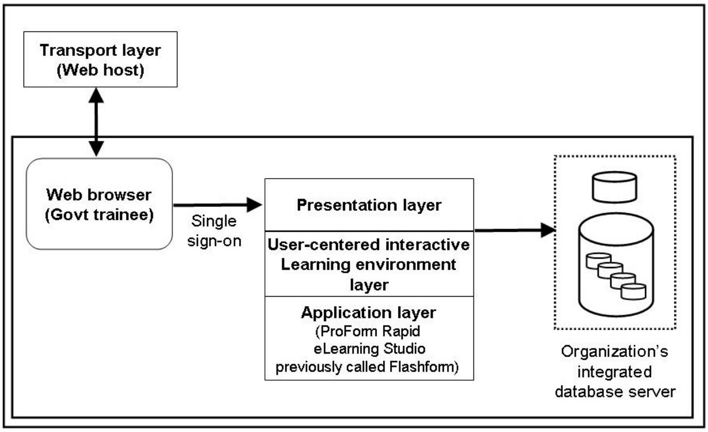


Fig. 17.4 The AELS architecture

The transport layer is the Web-host providing the Web-service. Once the trainee/user accesses the Web-host by entering the URL (e.g.: http://www.goal.edu.gov) the training content of the default page in the presentation layer (such as index.html, or default.php) is transferred from the Web-host's server for this site to the trainee's local computer and then displayed by their Web-browser. The training-program's content material served to the trainee/user is retrieved from the P-layer.

The presentation layer is invoked through the GUI of the JOOMLA CMS. The CMS provides the interface between the trainee/user and the IS where the necessary information (user details, departmental profile, and other information can be accessed).

The learning environment layer is the MOODLE LMS which is invoked through an interactive button from the homepage of the CMS. The LMS manages the learning activities of the users (training facilitators and trainees) as well as the LMS system administrator's tasks such as configuration of the system and giving access rights to trainees/users.

The application layer is the SCORM-compliant learning object or a packaged re-purposed Web-mediated training program. These materials are created with the use of a popular authoring tool such as ProForm Rapid eLearning Studio (previously called Flashform). An open-source authoring tool, e.g.: 'eXe' (http://exelearning.org/) can also be used as an alternative option. With eXe, authored learning materials can be seamlessly uploaded in MOODLE. Other software may be used to create more complicated learning objects (such as an avatar or 3D objects).

To conFig. The CMS and LMS environments, the LAMP environment will be used for the server and database components. We suggest that the LAMP environment affords the scalability and economic aspects of our AELS architecture, which is more realistic than the Alonso´s model. Ideally, an SSO invocation makes the integrated system accessible through the CMS login facility. Others have investigated this aspect (see http://www.opengroup.org/security/sso/). We implement this configuration and, simply call it ‘LAMP●’. Such kind of configuration is sketched in Fig. 17.5 as follows.

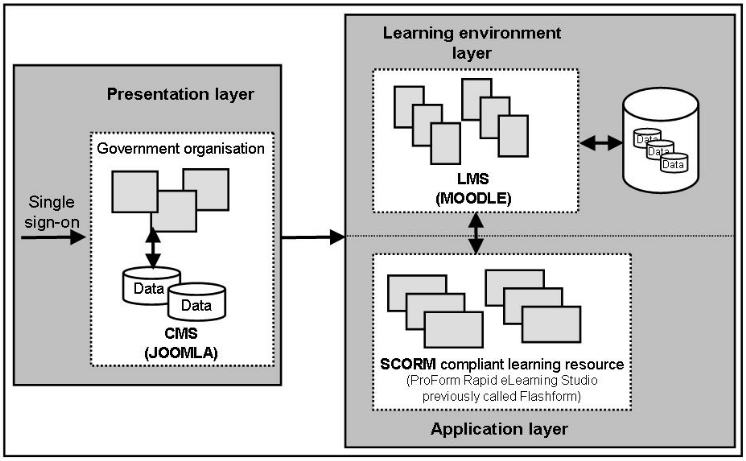


Fig. 17.5 LAMP● configuration for CMS-LMS platforms

17.4.3 Understanding the Need for Knowledge Navigation

This system has been designed as a tutoring system for *do it yourself* (DIY) Web-course creators or home-grown online course developers. This repurposing AELS is a user-centered computer-based, interactive, multimedia extension of the more familiar instructional tools used in face-to-face training sessions, such as textbooks and training manuals.

We find through our research that often CBT courseware tries to promote new and unfamiliar forms of learning that totally miss user-centered opportunities for learning. Moreover, these eResources may not fit established learning patterns. As such they are difficult for both instructors and learners to use.

Instead, textbooks and training manuals are common instructional tools that are used every day in thousands of educational and corporate sector training environments. In the corporate sector, attending training/reskilling sessions has become a time/cost dilemma; carefully placed technology training tools can be employed to enhance employees’ skills development experiences. Combining instructional strategies that promote effective HCI with timely learner guidance, multimedia elaboration and appropriate practice is proving to be effective in the acquisition of the corporate sector’s employee knowledge and skill advancement. Often in business the most effective use of employees’ computer activity occurs with spreadsheets, word processing and database management. There can be no doubt that these software packages have become popular everyday tools which significantly increase productivity and effectiveness.

Our user-centered AELS relies upon sound principles of instructional design to ensure the design of customized and instruction-centered, interactive, multimedia courseware for more cost effective training than is possible with other computer-based instructional development tools. The user-centered AELS assists participants in selecting the right skills building lesson. The skills building lesson steps included in our AELS cover much of the knowledge and skill taught in training programs. In many cases a given topic may have each of the types of skills building lessons available. Most topics include information for all the lesson types. However, the instructional goals usually dictate which of the skills building lesson types should be included for a given topic.

17.4.3.1 Knowledge Navigation Framework

In our AELS there are four types of interactive lessons that involve: just the *facts*, *parts of* things, *how* something works, *kinds of* similar examples (and non-examples), *try-me activities* (with feedback) and a final *interactive quiz* (that allows re-entry for follow up activities if necessary). This user-centered approach to learning is adapted from David Merrill's well-known concept learning strategies (Merrill et al. 1992), the following descriptions provide an overview of the instructional strategies of these lessons:

1. Facts: The need to know about a topic usually indicates a *fact* lesson. This includes knowledge about devices, people, places, creatures, activities, processes, etc. A *fact* lesson can be used for almost any topic and provides the building blocks for other learning,
2. Parts of: The *parts of* lessons are useful when knowing the parts of something or the location of something. A *parts of* lesson is appropriate for learning new terminology: learning places; learning controls of advice; how things go together; learning parts of a form, diagram chart, text format, etc. *Parts of* lessons are almost always necessary before learning a procedure or process,
3. How to: The need to know *how* something works or *how to* do something indicates the design of an appropriate *how to* lesson. For instance: for learning how to operate a device, how to assemble or disassemble something, how to interact with other people, or any other procedure. A *how to* lesson is appropriate for learning what happens in a natural process or how a device works,
4. Kind of: Identifying a new object, activity, or process as something already known indicates a *kind of* lesson. A *kind of* lesson is appropriate for comparing similar devices, activities, processes, or jobs. A *kind of* lesson is appropriate for recognizing the variety that can occur within a given class of things. Often neglected, *kind of* lessons are perhaps one of the most important types of learning. The ability to transfer (use) knowledge and skill to a new situation requires a *kind of* lesson in order for the learner to generalize. At the same time, the ability to recognize when a given activity or procedure may not be appropriate in a new situation requires a *kind of* lesson in order to learn to discriminate.

17.5 Pilot System Testing Program

The authors of this chapter conducted a pilot study to test our user-centered AELS courseware development tool and validate our research methodology for further experimental study. It was our intention to implement three training modes for methodological testing purposes (face-to-face only – *Treatment 1*, eLearning only – *Treatment 2*, and blended face-to-face and eLearning – *Treatment 3*). Using the area of government training, this was intended to address four main issues relating to employee reskilling:

1. Identify current in-house training practices, workforce needs, and intended learning content,
2. Design and development of adaptable, flexible eLearning tools,
3. Implementation of a research design to collect data from government trainees before and after implementing three training strategies (face-to-face only, eLearning only, and blended face-to-face and eLearning),
4. Analyze and interpret the results and report.

In order to test whether our TPLA model worked, the pilot study implemented a user-centered AELS built by a Web-development team that followed our courseware design model as the eLearning environment (*Treatment 2*). The learning content was chosen by one of our industry sponsors (see Acknowledgements). The organization held focus groups to determine a training topic. It was decided that 'an introduction to ethics' was widely applicable across government and important in maintaining the trust and respect of the public. The research team requested a total number of 75 government agency participants (20+ per learning strategy) and the industry sponsor sought to obtain the number required.

In checking the details of the research design before implementing the main study we were checking the internal consistency of our ethics knowledge pre-test and post-test by calibrating the test performance on each test so that learning outcomes could be measured on a common scale and checking the differentiating features of each learning strategy for effectiveness.

17.5.1 Instrument to Assess Cognitive Preference

The instrument chosen to assess cognitive preferences was the *Object-Spatial Imager and Verbal Questionnaire* (OSIVQ) by Blazhenkova and Kozhevnikov (2009). This instrument is a paper-based self-report questionnaire designed to capture people's preferences to use mental imagery versus verbal representations. Blazhenkova and Kozhevnikov (2009) maintain that mental imagery is the ability to mentally represent things in the mind (especially visible objects), not by direct perception but by memory or imagination. This cognitive preference measurement instrument employed a five point scale using "5" rating to indicate absolute agreement and "1" to indicate total disagreement. The instructions to the participants included an assurance to them there were no right or wrong answers; instead asking for them to provide honest responses to the questions posed.

To ensure that this instrument would be suitable, the team gathered responses from a trial group of colleagues and ran a preliminary trial of the analyses before using the instrument in the pilot study. This preliminary trial provided information about the internal consistency of the three separate scales produced by the OSIVQ.

17.5.2 Trial

17.5.2.1 Evaluate Prior Domain Knowledge of Each Participant

Participants received a plain language statement as an eMail attachment before the session and signed a consent form. Participants were directed to the main classroom, set up with tables and chairs in a traditional workshop *'U'* shape.

A short introductory/welcoming session was given by the researchers, briefly outlining the research to put participants at ease in a general sense as well as to reaffirm their anonymity so far as the researchers were concerned.

We managed the paper-based open-ended pre-test, as described above. To further align the focus group data with the pilot study data, three categories of information in the focus group feedback were also sought from the pilot study participants. This categorical data was to be conveyed from the pilot study participants back to the research team by each participant entering a response to the focus groups' categorization data in a set of small tick-boxes placed at the top of the pre-test. The informational categories were: preferred training mode, experience with eLearning, should in-service learning for work purposes by any mode be considered part of one's workload. Responses contributed to the subsequent analysis.

17.5.2.2 Collect Information about Cognitive Preference for Each Participant

Each participant was given the OSIVQ; and it was anticipated that OSIVQ three scale scores was used to split them into two treatment groups, according to cognitive preference.

Our cognitive measurement expert, Professor Izard, coded the OSVIQ responses in preparation for analysis using the Quest: Interactive Test-Item Analysis system. However, time restraints meant this task could not be completed without delaying the remainder of the pilot study, so random allocation was used instead.

17.5.2.3 Deliver the Learning Content Area's Training Approaches

Group 1 (12 participants) received T*reatment 1* (face-to-face) in the initial classroom while group 2 (11 participants) were allocated to T*reatment 2* (eLearning) in the adjacent dedicated computer laboratory.

It was important to keep both treatment groups separate at this point of the Pilot Study. This separation was necessary to maintain throughout the process, including tea and coffee break. Both groups were combined in the main classroom for the post-test.

17.5.2.4 Evaluate Post-Training Knowledge on the Same Knowledge Continuum

At the end of the training all participants undertook the open-ended post-test together in the main treatment room. Part A involved five sections with a total of 13 separate parts – some of which were the same as the questions on the pre-test, and a Part B of the post-test which involved the four case studies.

The scoring rubric for the post-test provided for 29 separate sub-scores, one for each part of an item and for each of the four case studies.

17.5.3 Statistical Outcomes

17.5.3.1 Preparing the Data for Analysis

Each item part on the pre- and post-test papers was first scored by the research team during October-November, 2010, and cross-checked to ensure that all correct answers received credit, regardless of their wording. Results for all participants were coded in an Excel spreadsheet by Izard.

17.5.3.2 Subsequent Analysis and Interpretation

Pre- and post-test results were analyzed with an item matrix that had each individual's responses for every item part. Common items (identically worded) were anchored so that scale scores on the pre-test were comparable with scale scores on the post-test.

The difference between pre- and post-test scores indicated whether learning occurred, whether no learning occurred, or whether the instructional strategy resulted in reduced achievement.

17.5.3.3 Results

The 12 participants in *Treatment 1* (ran out of time) had a pre-test mean achievement scale score of -0.80 with a standard deviation of 0.93. The corresponding post-test mean achievement scale score for the same 12 individuals in *Treatment 1* had a mean achievement scale score of -0.27 with a standard deviation of 0.34. In terms of mean learning achievement, the *Treatment 1* group moved up from -0.80 to -0.27, a gain of 0.52 scale units (when rounded).

The corresponding 11 participants in *Treatment 2* (finished earlier than *Treatment 1*) had a pre-test mean achievement scale score of -0.17 with a standard deviation of 0.75. The corresponding post-test mean achievement scale score for the same 11 individuals in *Treatment 2* had a mean achievement scale score of -0.10 with a standard deviation of 0.57.

In terms of mean learning achievement, the *Treatment 2* group moved up from -0.17 to -0.10, a gain of 0.07 scale units (when rounded). These changes are minimal. Ways of addressing this problem are discussed in the next section.

17.6 Discussion

The expectation of substantial gains with a short training period around two hours is unrealistic for both Treatments 1 and 2 (and by extension, presumably, Treatment 3). Future investigations should include 1-day, 2-day and 2+day training to infer the duration of training that will allow substantive magnitudes of learning to be detected. Similarly, the size of each training group needs to be larger: it is difficult to justify such small groups being involved in training given the costs associated with providing trainers, the provision of suitable facilities, and the transport costs for both presenters and participants.

Face-to-face learning can provide more opportunities for feedback to participants. It would be better to add further feedback opportunities to the AELS instructional framework to ensure there is greater control of the magnitude of feedback which may be an alternative explanation of differential learning.

Secondly, limiting the learning area to a single content area (such as '*ethics*') would provide no evidence of the extent to which the information obtained generalizes to other learning content areas. Additional content areas need to be added, with sufficient time allowed for the university research team and the industry sponsors to generate eLearning content and pre- and post-tests.

The ethics pre-test has an internal consistency of 0.61 without deleting or adding test-items. This value needs to be improved by using the analysis to modify some test-items, and perhaps to move some test-items from the post-test to the pre-test so both tests are of similar accuracy.

The ethics post-test has an internal consistency of 0.77 without deleting any test-items. Refinement of this test by deleting Part B test-items that failed to detect any differences between participants with knowledge and participants lacking knowledge on the dimension of interest (applied knowledge of ethics), would serve to improve the evaluation of post-training knowledge. These procedures will be followed in the next analyses while waiting for advice from *Government Skills Australia* (GSA) on the groups to participate in the study. The complexity of the case studies used in the face-to-face group (*Treatment 1*) may well be affecting the post-test evidence of achievement. This shows when subjects identify stakeholders (for example), but do not choose the best solution or explain why that is so.

The OSIVQ instrument does not have similar internal consistency indices for each of the three scales. This suggests that, either permission has to be obtained to extend the weak scale, or an alternative cognitive preference instrument should be chosen to replace the OSIVQ. The scoring of the OSIVQ needs simplification to provide this information efficiently.

Given the complexity of bringing together all the necessary pieces of the online training/learning environment puzzle, how does corporate management make their decisions to implement a technological solution to solve their business training needs? In the recent past, academe and corporate decision makers may have been forced to take different approaches (Dick et al. 2006). The trouble is there is no blue print for anyone to follow.

During the last decade the Web has given the educational technologists among us a completely new environment. This ICT led revolution has fundamentally changed the practice of distance learning and teaching (Anderson 2008). Therefore, the corporate-level decisions to adopt sensible technology-based training solutions may have already bolted; the lines between academe and the corporate training environments are quite blurry. The evidence is all around us. There has been a major transformation of the single-mode distance universities to leverage themselves in a cost effective marketplace. Individuals can now cherry pick their training opportunities.

In a sense, a skilled work-force has become a portable commodity. Consequently, the *chief information officer* (CIO) in the business environment must wrestle with the dilemma posed by this emerging open access to training opportunities afforded by the Web. As Dick et al. (2006:896) identify: *"training or skills and knowledge acquisition is expensive"*. In business there are decisions that involve larger issues than the procurement of the most suitable LMS. Things like where to locate the technical infrastructure, and when is it the best time to remove people from their workplace to participate in training. More often than not these days, there is a growing expectation – that employees undergo their *professional development* in their own time.

In seeking out reasons for ICT adoption through the lens of a trainee, the rhetoric on the virtues of online learning is long and well covered in the literature (Anderson 2008). Similarly, an examination of what people do with their online learning experiences would be a daunting task beyond the scope of this chapter. Suffice to add that online learning takes place regardless of time zones, ignoring the tyranny of distance. The future for Web-mediated learning is bright, according to Schank (2002: xvi): "Despite the shortsightedness of some executives, the future of e-learning is so bright that we may all have to wear shades". It comes down to if the training materials are fun, providing for failure and rebirth all at the same time. Schank (2002) provides a set of powerful design and delivery principles that guide the corporate sector to build more effective training solutions. These principles amount to the next that: feelings count; people are not empty jars; they choose the right time; to throw them in the deep end; avoid the white knight syndrome – self taught sticks best; to rule out rote learning; allow for the square pet in a round hole; to make an entrance; to use practicing experts and train the many.

17.7 Conclusion

The main purpose of this chapter is to outline a user-centered AELS design model that is easy to reproduce as a seamless Web-mediated courseware development tool. Along the way, we have shown how some of the technical IS barriers may appear too difficult to overcome for novice Web-content developers. These issues may explain why the adoption rate of ICT tools in corporate training programs is not keeping pace with technological advances.
Far too often plans for the *proof of concept* or prototype IS are drawn up by a non-technical training developer, with specifications that ignore the complexities pertaining to the expected functionality (McKay and Martin 2007).

We were able to show that our AELS courseware development tool was effective in so far as it provided an effective and seamless online learning environment.

We were able to ascertain that the participants in T*reatment 3* (eLearning only instruction) were able to demonstrate a better increase in their knowledge and skills associated with making ethical decisions than the participants who were in T*reatment 1* (face-to-face only instruction).

As such, we were able to show that our TPLA design model combines Web-hosting service access for trainees in a seamless process, providing employees training modules on an 'as needed' basis to afford individual choices as they undertake training programs.

Our system architecture means that each IS component can be designed to maximize the interoperability between the functional TPLA layers. Unlike other LMS architectures where the corporate or training manager has the ability to elect to track a learner/trainee and control what they see and do (Dick et al. 2006), the TPLA design layers provide for complete anonymity to be *built* into a training session (only at the behest of the IS administration and only if the IS implemented supports it). Furthermore, the separation of the TPLA layers allows for a user-centered Web-mediated training IS to be designed such that the *granular level* instructional strategies (quality control) are easily maintained. Another barrier to effective online adoption practice relates to the lack of instructional specifications for quality control of learning/training content development.

We propose that our TPLA cuts down the hidden technology stumbling blocks of the past by providing *stateful processes* that facilitate user-centered approach to learning. The autonomous nature of the TPLA design layers reflects the *nondeterminism* as described by Marconi and Pistore (2009:19), which simply means our model can deal with the external IS services that cannot always be taken for granted. The literature reports that more work is constantly underway to improve Web-services. We ascertain that our user-centered TPLA extends the reach of IS design.

SCORM compliance is often perceived as a bothersome online development issue for Web-developers, novice and expert alike (Pohl et al. 2009). Our TPLA application layer neatly takes care of this mandatory requirement; when the courseware system is developed in a packaged tool such as *ProForm Rapid eLearning Studio* (or a similar off-the-shelf templating development application). Yet it is the SSO that serves to innovate the TPLA (The Open Group 2008). The quest for an SSO is the feature which aggravates users the most; they expect seamless integration and interoperability when visiting Web-sites (Cohen 2002), thus far the real world is still awaiting its arrival.

Finally, while our pilot study tested our user-centered AELS courseware development tool and validated our research methodology, we still have more work to do. It was our intention to implement three training modes for testing purposes (face-to-face only, eLearning only, and blended face-to-face and eLearning).

The preliminary results revealed that it was unrealistic to expect to identify the interactive effect of learning preference and instructional format (the computerized delivery) in the allotted two hour timeframe. To overcome these issues we plan further research experiments with larger cohorts of participants and with different timeframes to allow us to identify substantive magnitudes of learning and further improve the user-centered approach our AELS model has for online learning module development.

Acknowledgments. This research was funded by the Australian Research Council (ARC) Industry Linkage Project (LP0989245) and our industry sponsors were: Government Skills Australia (GSA) and NetEffective Media Group. We wish to thank Rilke Muir for crafting our technical jargon to appeal to a wide range of readers (http://www.truewordsdivined.com.au/).

References

Alonso, G., Casati, F., Kuno, H., Machiraju, V.: Web services: Concepts, architectures and applications. Springer, CA (2004)

Anderson, T. (ed.): Theory and practice of online learning, Athabasca University (2008), `http://www.aupress.ca/index.php/books/120146` (accessed November 5, 2011)

Anderson, N., Lankshear, C., Courtney, L., Timms, C.: Girls and ICT survey: Initial findings, Curriculum Leadership Journal, `http://Cmslive.Curriculum.Edu.Au/Leader/Default.Asp?Id=13812` (accessed November 5, 2007)

Blazhenkova, O., Kozhevnikov, M.: The new object-spatial-verbal cognitive style model: Theory and measurement. Applied Cognitive Psychology 23(5), 638–663 (2009)

Clark, R.C., Kwinn, A.: The new virtual classroom: Evidence-based guidelines for synchronous e-Learning. Wiley, CA (2007)

Cohen, F.: Using Web services for e-Commerce sign-in: SOAP authentication for distributed computers (2002), `http://ibm.com/developerworks/webservices/library/ws-single/` (accessed November 5, 2011)

Dasgupta, S., Granger, M., McGarry, N.: User acceptance of e-collaboration technology: An extension of the technology acceptance model. In: Group Decision and Negotiation, vol. 11, pp. 87–100. Kluwer, The Netherlands (2002)

Castro, E.: HTML for the world wide web: Visual quickstart guide, 4th edn. Peachpit Press, CA (2000)

Corbitt, A.T., Koedinger, K.R., Anderson, J.R.: Intelligent tutoring systems. In: Helander, M., Landauer, T.K., Prabhu, P.V. (eds.) Handbook of human-computer interaction, 2nd edn., pp. 849–874. Elsevier Science B.V., Amsterdam (1997)

Dick, G.N., Case, T.L., Ruhlman, P., Van Slyke, C., Winston, M.: On-line learning in the business environment. Communications of the Association for Information Systems 17(41), 895–904 (2006)

Driscoll, M.M.: Developing synchronous web-based training for adults in the workplace. In: Khan, B.H. (ed.) Web-Based Training, pp. 173–183. Educ. Tech. Publication, NJ (2001)

Forbus, K.D., Feltovich, P.J.: Smart Machines in Education. MIT Press, UK (2001)

Gery, G.: Making CBT happen: Prescriptions for successful implementation of computer-based training in your organization. Harper & Row, NY (1987)

Jacobs, I., Walsh, N.(eds.): Architecture of the world wide web, 1, `http://www.w3.org/TR/webarch/` (accessed November 5, 2011)

Jameson, A.: Adaptive interfaces. In: Sears, A., Jacko, J.A. (eds.) The Human-Computer Interaction Handbook: Fundamentals, Evolving Technologies, and Emerging Applications, pp. 433–458. CRC Press, NY (2008)

McKay, E.: Instructional Strategies Integrating the Cognitive Style Construct: A Meta-Knowledge Processing Model (Contextual Components That Facilitate Spatial/Logical Task Performance). PhD Dissertation, Deakin University, http://tux.lib.deakin.edu.au/adt-VDU/public/adt-VDU20061011.122556/ (accessed November 5, 2011)

McKay, E.: Planning effective HCI to enhance access to educational applications. International Journal Universal Access in the Information Society 6(1), 77–85 (2007)

McKay, E.: The human-dimensions of human-computer interaction: Balancing the HCI equation. IOS Press, Amsterdam (2008)

McKay, E., Axmann, M., Banjanin, N., Howat, A.: Towards web-mediated learning reinforcement: Rewards for online mentoring through effective human-computer interaction. In: Proceedings of IASTED (2007)

McKay, E., Martin, J.: Multidisciplinary collaboration to unravel expert knowledge: Designing for effective human-computer interaction. In: Keppell, M. (ed.) Instructional Design: Case Studies in Communities of Practice, pp. 309–329. Idea Group, UK (2009)

McKay, E., Merrill, M.D.: Cognitive skill and Web-based educational systems. In: McKay, E. (ed.) eLearning Conference on Design and Development: Instructional Design - Applying first principles of instruction. Australasian Publications On-Line, Informit Library (2003), http://www.informit.com.au/library/96-108 (accessed November 5, 2011)

Merrill, M.D.: Pebble-in-the-pond model for instructional development. Performance Measurement (2002), http://www.ispi.org/pdf/Merrill.pdf (accessed November 5, 2011)

Merrill, M.D., Tennyson, R.D., Posey, L.O.: Teaching concepts: An instructional design guide, 2nd edn. Educational Technology Publications, New Jersey (1992)

Peng, H., Chou, C., Chang, C.-Y.: From virtual environments to physical environments: Exploring interactivity in ubiquitous-learning systems. Educational Technology & Society 11(2), 54–66 (2008)

Pohl, H.-M., Deicke, B., Milde, J.-T.: From Paper to Module – An Integrated Environment for Generating SCORM Compliant Moodle Courses Out of Text and Multimedia Elements. In: Jacko, J.A. (ed.) HCI International 2009, Part IV. LNCS, vol. 5613, pp. 196–203. Springer, Heidelberg (2009)

Schank, R.C.: Designing world-class e-Learning: How IBM, GE, Harvard Business School, & Columbia University are succeeding at e-learning. McGraw-Hill, New York (2002)

The Open Group. Introduction to single-sign-on (2008), http://www.opengroup.org/security/sso/ (accessed November 5, 2011)

Zapata-Rivera, J.-D., Greer, J.: Inspectable Bayesian student modelling servers in multi-agent tutoring systems. International Journal of Human-Computer Studies 61(4), 535–563 (2004)

Abbreviations

AELS	Adaptive Educational-Learning Systems
AI	Artificial Intelligence
ARC	Australian Research Council
ASCII	American Standard Code for Information Interchange
CAL	Computer Aided Learning

CBT	Computer-Based Training
CIO	Chief Information Officer
CMS	Content Management System
DMSA	Dreyfus Model of Skill Acquisition
DIY	Do It Yourself
ELS	Educational-Learning Systems
FACS	Fellow of the Australian Computer Society
GSA	Government Skills Australia
GUI	Graphical User Interface
HCI	Human-Computer Interaction
HTTP	HyperText Transfer Protocol
ICT	Information Communication Technology
IS	Information Systems
LAMP	Linux-Apache-MySQL-PHP
LMS	Learning Management System
OSIVQ	Object-Spatial Imager And Verbal Questionnaire
PhD	Doctor of Philosophy
RMIT	Royal Melbourne Institute of Technology University
SCORM	Sharable Content Object Reference Model
SSO	Single-Sign-On
TPLA	Transport, Presentation, Learning Environment and Application
ULS	Ubiquitous-Learning System
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
W3C	World Wide Web Consortium
XHTML	eXtensible HyperText Markup Language
XML	eXtensible Markup Language

Chapter 18
Intuitionistic Fuzzy Logic-Based Approach of Intrinsic Motivation in CSCL Settings during Illusionary Sense of Control

Sofia Hadjileontiadou[1], Georgia Nikolaidou[2], and Leontios Hadjileontiadis[3]

[1] Hellenic Open University
Praxitelous 23, 10562 Athens, Greece
shadji@ee.duth.gr
[2] Direction of Primary Education of Eastern Thessaloniki
Katsimidi-Milou, 54638 Thessaloniki, Greece
msa0848@uom.gr
[3] Department of Electrical & Computer Engineering, AUTH
University Campus, 54124 Thessaloniki, Greece
leontios@auth.gr

Abstract. We examine if the provision of illusionary sense of control that resides in the collaborators is perceived as actual control and cause intrinsic motivation towards better work within a Computer Supported Collaborative Learning (CSCL) environment. On the basis of expert knowledge, indicators are set to support intelligent decision making upon the quality of the collaboration. The fuzzy logic translates knowledge from the qualitative to the quantitative plane. It supports a model of indicators on the basis of the expert knowledge that is expressed through linguistic descriptions. However, the qualitative nature of such decisions, entail some uncertainty and hesitancy during the modeling procedure of the specific knowledge. The Intuitionistic Fuzzy Logic (IFL) enables the capture and expression of this uncertainty and hesitancy while modeling the decision making upon the quality of the collaboration, thus it extends the fuzzy logic possibilities. The efficiency of an IFL-based approach is explored through a case-study targeting the increase in the quality of the collaborative performance. The approach leads to a more minimal design of the supporting mechanisms in CSCL settings.

18.1 Introduction

According to the self-determination theory (Deci and Ryan 2000), learner's control is assumed to enhance intrinsic motivation, since it fulfills the need and promotes his/her sense of autonomy (Katz and Assor 2007). From this theoretical point of view, giving control to the learner is believed to enhance motivation.

A. Peña-Ayala (Ed.): Intelligent and Adaptive ELS, SIST 17, pp. 443–468.
springerlink.com

Moreover, it is pursue to have positive effects on learning outcomes (Kinzie and Sullivan 1989; Williams 1996).There are a number of learner's characteristics that influence the relation between learning control and learning outcomes, some examples being prior knowledge, ability, motivation and perceptions about the learning process (Williams 1996). In any case, however, learner's control can be considered as adaptivity defined by the learner him/herself.

Recent research stressed the importance of perception as a supplementary factor influencing the effectiveness of learner's control (Elen and Clarebout 2006). (Katz and Assor 2003) states that “Tasks that are consistent with the learners’ individual interests strongly enhance their sense of autonomy and, therefore, enhance intrinsic motivation”. Even though the offered choices are illusory or it seems trivial, choice becomes meaningful if the learner experiences it as such (Katz and Assor 2007). The perception of control has been mentioned as a critical variable at the beginning of a predictive chain, involving both achievement and motivation (Harter and Connell 1984). Perceived control, is therefore, suggested as a precursor to both achievement and motivation (Kinzie 1990). In this vein, simply embedding control or choice options in a learning environment may not be sufficient. If a learner only has control over features that do not seem relevant to the learner, or if learners do not see the choices from which they can choose as sufficiently different, then they do not perceive any control and, hence, learner's control might not achieve the intended effect (Katz and Assor 2007).

The degree of learner's control does not always need to be very advanced. More basic levels of control, such as control over the pace of learning, can already give the learners a sense of autonomy or ownership of their learning process (Dror 2008). Even if control is illusionary (such as giving learners the feeling of control when in fact they do not have any control) this can provide a means for improving learning outcomes (Dror 2008).

In line with the above, this chapter introduces a new framework that places the concept of learners' intrinsic motivation under an illusionary sense of control within a CSCL setting. Boulding's typology (Boulding 1956, Gabriele 2010) is adopted to explicitly describes the structure and function of subsystems of the collaborative setting and express the way control externalizes intrinsic motivation through the firing of collaborative interactions. Efficient modeling of the latter provides quantitative measures of the quality of peers' collaboration, as a means to monitor the learner's adaptivity during his/her engagement within the CSCL setting.

The proposed framework is supported by the experimental results from a case study referring to peers' collaborative construction of concept maps within a CSCL setting, showing effective contribution to the illusionary sense of control in the enhancement of peers' quality of collaboration, cognitive structure and learning.

18.2 Computer-Supported Collaborative Learning Settings

18.2.1 The Concept of Collaboration

In the research literature there is a broad acceptance of what comes under the umbrella of "collaboration". As Wulf (Wulf 1993) points out: "The word *collaboratory* suggests an amalgam of the words collaboration and laboratory". In Wulf's context the idea of collaboration is based on scientific inquiry supported by information technology. In addition, collaboration was a commonly used word in the Social Sciences, Education, Humanities, business, and communication literature.

There are several different types of collaboration identified in that literature, but most describe *a group of individuals bringing expertise from diverse disciplines to achieve a common goal* (Abramson and Mizrahi 1996, Green and Etheridge 1999, Berg-Weger and Schneider 1998, Walther-Thomas et al. 1999, Holmes and Howson 2000). In the area of education, the term *collaboration* has been applied to learning activities and provides a key point for learning theories.

For many years, theories of collaborative learning tended to focus on how individuals function in a group (Dillenbourg et al. 1995). More recently, the focus has shifted so that the group itself has become the unit of analysis. Therefore, nowadays, researchers address two different issues. The first one is based on the individual and the understanding of how one cognitive system is transformed by messages received from another. The second examines the group and how these cognitive systems merge to produce a shared understanding of problem solving (Dillenbourg et al. 1995).

Based on such perspectives, we can distinguish three different theoretical approaches to collaborative learning (Dillenbourg et al. 1995): the socio-constructivist; the socio-cultural and the shared cognition.

The socio-constructivist approach emphasizes the role of interactions with others rather than the actions themselves, revealing the way individual cognitive development is affected by social interactions. In this way, individual cognitive development follows a spiral of causality, that is, it allows participation in certain social interactions, which produce new individual states that, in turn, trigger more sophisticated social interactions, and so on (Dillenbourg et al. 1995).

The socio-cultural approach involves socially structured activities. Within the classroom settings, children take part in these activities, engaging with other peers (classmates) and adults (teachers). Through their collaboration with others, within the learning context, they deploy their understanding, which derives from the process of learning and as Wertsch (Wertsch and Stone 1979, p. 21) argues: "The process is the product". In this process, pupils and their social partners are independent and have active and dynamically changing roles (Rogoff 1995). In fact, the specific processes that occur when they communicate and share in order to make a decision are the body of cognitive development (Dillenbourg et al. 1995).

The shared cognition differs from the socio-constructivist and socio-cultural approaches, since the latter are concerned with the inter-individual plane, whereas the former focuses on the social plane, where emergent conceptions are seen as a group product (Dillenbourg et al. 1995).

18.2.2 Collaborative Interactions

Collaboration could also be defined as the *collaborative interactions* of learners. In fact, collaboration could be seen as a situation in which the learners interact in a collaborative way. These interactions, in order to be characterised as collaborative ones, involve *interactivity*, *synchronicity* and *negotiability* (Dillenbourg 1999).

In order to examine *interactivity*, closer attention must be given to the extent to which interactions influence the peers' cognitive processes (Dillenbourg et al. 1999). The second criterion, *synchronicity* implies the action of 'doing something together' and can be defined as a *considerate meta-communicative contract*, when Peer-*A* as a speaker, expects from Peer-*B* as a listener, to wait for his/her message and process the message as soon as it is delivered (Dillenbourg et al. 1999). As far as *negotiability* is concerned, group members are not traditional imparters of knowledge but they make allowances for each other for arguing, justifying, explaining and attempting to convince.

Focusing on the task level, participants have the chance to negotiate and examine several patterns of their conversation. For instance, sometimes they negotiate:

- How to interact (e.g., "Is that a question or a claim?"),
- To tune the turn-taking (e.g., "Let me speak") developing, hence, a meta-communication process (Dillenbourg et al. 1999).

It could be argued that a significant range of collaborative interactions can be deployed during the task problem by peers who build explanations, justify themselves or reformulate statements. The context of collaborative learning embodies a variety of social exchanges, interactions and their process, which affect the individual's performance and it would be valuable whether: "[…] researchers no longer (or should not) treat collaboration as a 'black box', but *zoom in* on the collaborative interactions in order to gain better understanding of the underlying mechanisms" (Dillenbourg 1999, p. 17).

In other words, it would be valuable to know whether a more complete analysis of the social context of peer interactions based on a collaborative learning environment could take place with the aim of developing an understanding of the complex processes that reciprocally affect each other.

In CSCL settings, where computer mediated collaboration takes place, log files of the collaborative interactions provide empirical raw data that carry information relevant to the collaborative activity. Elaborations of these data contribute to analysis of this activity therefore deepening our understanding of the complex processes it entails i.e., *zoom in* on it.

Methodological approaches towards the analysis of the collaborative interactions range from qualitative analyses of small interaction-rich episodes of collaboration, to quantitative measures of suitably categorized events of interaction that are indicative of the success of collaboration in some of its facets (Stahl et al. 2006). The quantitative formalization of the evaluation outcomes in both approaches, in the form of indicators, facilitates communication and mutual understanding of the modeled parameters of the collaborative activity. Moreover, within the CSCL settings, these indicators constitute a tool to dynamically follow aspects of the individual collaborative performance and upon established norms provide adaptive support to each learner (i.e., on the basis of his/her collaborative activity).

18.3 Indicators of the Quality of Collaboration

Collaboration means to *co-labour*, to work together. It does not necessarily mean harmony or complete agreement with someone, but it does mean having a working relationship where shared interests are served through the processes of dialogue and cooperation. In collaboration, person's perspective does not dominates. Instead, a perspective emerges through dialogue that neither person would have had independently. Hence, collaborative relationships are inherently creative. In CSCL this effort is reflected in a series of interactions that the collaborators make in the common space, in the form of entities. Upon each entity each collaborator can perform actions that reflect his/her contribution to the collaboration. Upon analysis of these entities and actions, indicators of the quality of the collaboration (QoC) are set, but within a multidimensional conceptual framework of collaboration where broad aspects of collaboration are realized and the QoC are set (Meier et al. 2007).

18.3.1 Balance

Assertiveness and listening are complementary aspects of interpersonal *communication.* Both are essential, in varying degrees, if someone is to succeed in balancing her/his own and others' needs, creating an atmosphere of effective collaboration and communication. Listening and asserting are not the same as hearing and speaking. Instead, they reflect which point of view is in someone's consciousness at any given moment. When someone listens to others, she/he is allowing their point of view to prevail in her/his awareness; when she/he asserts, her/his point of view prevails. To collaborate, she/he must balance the two perspectives in her/his awareness and allow the synergies from those competing views to emerge as insight. The justification or not of the latter assumption is definitely reflected in peers' dialogue during their collaborative activity. Moreover, peers' actions can be used as a means to find additional information about the existence or not of a balanced activity in the joint task.

Hadjileontiadou et al. 2003 introduced the area of balanced collaboration activity, where the quality of collaboration is maximised when peers converge on a balanced collaboration. They stated: “The idea of a balanced collaboration entails an effort from the more knowledgeable to help his/her collaborator to be more active (e.g. using questions, asking for clarifications and so on). So the core of the balanced collaboration activity (BCA) is the promotion of collaboration instead of individual work” (Hadjileontiadou et al. 2003, p. 319). An indicator of a BCA can be the proper management, by the collaborators, of the turn-taking mechanism (i.e., the turn of submitting either an entity or an action).

18.3.2 Cognitive and Productivity Rates

Another aspect of the collaboration conceptual framework is the *information processing* (Meier et al. 2007) that refers to the task of the collaboration. Through the exchange of information a shared meaning of the task is established. An entity carries information concerning the task and the *cognitive rate* (i.e., the rate of the submission of entities by the collaborator can be indicative of his/her contribution to the information processing).

Within this aspect, the *productivity rate* is set as another indicator of the QoC. In particular, it refers to the rate of the actions that are performed upon the entities and influence them as information carriers (e.g., by adding them, deleting them, altering their content). It is obvious that due to the multidimensional character of the collaboration conceptual framework the focus of the analysis of collaborative interactions is put upon important instances of collaboration as they are captured through the established indicators, in this case the *balance* and the *cognitive* and the *productivity* rate. *Motivation* is a core aspect of the conceptual framework of the QoC as it sustains dedication to the collaboration (Meier et al. 2007). In this work it is used as the basis to formulate the hypothesis: "*If motivation is triggered then a higher value of the QoC will be produced, measured on the basis of the three aforementioned indicators*".

18.4 Intrinsic Motivation through Illusionary Sense of Control

18.4.1 Boulding's Typology

Motivation lies at the individual plane, triggering it within CSCL leads to extend the conceptual framework of the collaboration, and regard it as a complex system. The Boulding's typology (Boulding 1956, Gabriele 2010) contributes to this effort by making explicit the structure and function of subsystems as shown in Fig. 18.1.

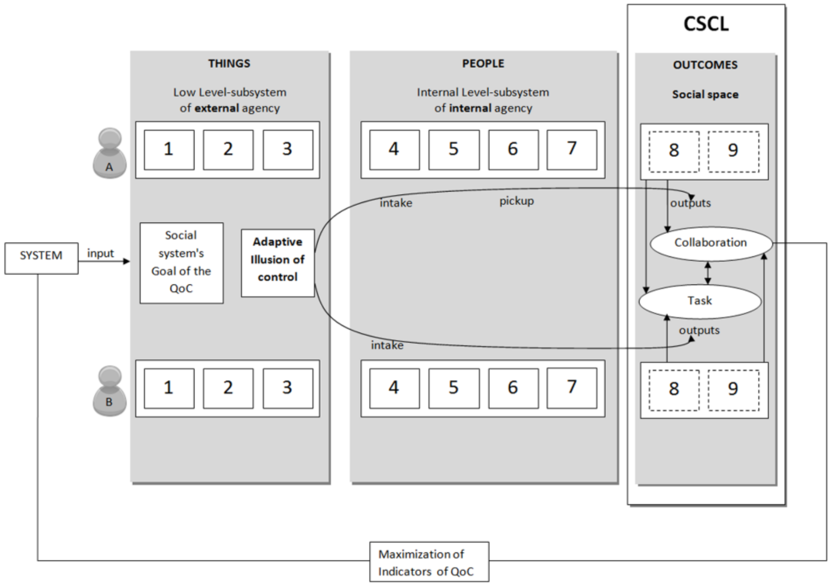


Fig. 18.1 Structure and function of subsystems of the collaborative setting according to Bouldings' typology (Boulding 1956, Gabriele 2010, Hadjileontiadou et al. 2011)

Within the collaborative setting, the things (levels 1-3) the people (levels 4-7) and the outcomes (levels 8, 9) can be realized. Levels 1-3 comprise spatial and temporal traits and information respectively (Checkland 1981). Examples of the first two levels in schools are formal goals, participant roles, equipment and rooms for frameworks, calendars, schedules and routine activities for clockworks (Gabriele 1997). Examples of the third level with the thermostat metaphor are evaluations of the recourses and needs and estimation of the workload in the classroom that leads to self-regulated remedy actions, if needed (Gabriele 1997).

The first three levels that comprise the things of the system are either externally (levels 1 and 2) or self-regulated (level 3) to externally prescribed inputs (Checkland 1981). Levels 4-7 comprise the people, who are self-regulated to internally prescribed criteria (attractors), thus they are externally undesignable.

In level 4, educational resources, information and programs should be available to the individual who is acting according to interior criterion (instinct). Thus, the input here is described as 'intake' (Gabriele 1997). Level 5 realizes the variability of the interior criterion due to the differences of the people (heredity), which is inherited the response to the stimulus.

Furthermore, in level 6, the individual perception intervenes between the stimulus and the response; hence, from this level on, the input/intake is described as 'pickup' (Gabriele 1997). The individual 'picks up' whatever she/he believes in or understands, otherwise his/her output will be distorted by his/her perception, (e.g., instead of learning, *memorizing*) (Gabriele 1997).

At level 7 the individual acts consciously to produce the output not only according to his/her perceptions of the 'pickup', but even upon new stimulus that she/he can create from the processing of his/her experiences. The 'people' levels are depicted by physical boundaries. At levels 8 and 9, where the social outcome is expected, only if the individual participation is voluntary and authentic will the social intangible system exist. Moreover, at these levels, individual goals are to be met before the social, otherwise the individual seeks his/her own (Gabriele 2010).

The Boulding's typology clarifies which levels (subsystems) are predictable and controllable within a system and which are not (fact 1), classifies the relevant types of external (inputs) or internal (attractors) stimuli for their function (fact 2), clarifies that when predictable and designable inputs are designed at levels 1-3, they are first expected to be perceived at level 6 and then reformed to attractors of intrinsic motivation at level 7, towards fulfilling initially individual and then social goals from level 8 and so on (fact 3) (Gabriele 2010).

Within the above CSCL context, the assistance dilemma arises (i.e., when to provide support and when to withhold it) in order to enhance learning (Kapur and Rummel 2009). This dilemma introduces issues concerning the timing, the visibility and the adaptivity of the support to be provided.

Timing may vary in a continuum between the decision of the provision of the support at the beginning of the task, and a delayed one. Visibility of the support may be connected to the timing of its provision and may have a gradually fading presence. When the timing and visibility are automatically decided by the system that mediates learning, according to the learner's needs, then the support that is provided is of an adaptive character. Moreover, the adaptation issues concerning the assistance dilemma may be extended to issues regarding the level of the stimuli provision and its type according to the Boulding's typology.

For example, when the effort to support the learner is put at level 7 (i.e., the collaborative performance as outcome), the ID may foresee the design of low-level inputs (i.e., designable 'things'), which will be expected to be reformed to internal attractors (i.e., of the collaborating 'people') that might maximize the opportunities to produce intrinsic motivation and better learning (i.e., better collaborative performance as 'outcome').

18.4.2 Sense of Control

Control provided to the learners within a learning setting serves as an input in the form of support set in Fig. 18.1. Dror 2008 states: "The learners' control can take many forms and can be viewed as a continuum". Control is totally surrendered to the learners, giving them full freedom to do (or not do) as they please. At the other extreme of the continuum, the learners have no control at all. Yet, if learners are not interested in or do not understand the choices of control provided to them (e.g., level 7 of the Boulding's typology), their involvement in learning may be decreased, which results in poorer learning outcomes (Paas et al. 2005).

Control that is interesting and understood by the learner is therefore suggested as a precursor to both achievement and motivation (Kinzie 1990). Even more, when the offered control is illusionary, choice becomes meaningful if the learner perceives it as such (Corbalan et al. 2009). On the other hand, if choices concerning control are too many, this may cause excessive cognitive load and become detrimental to learning (Dror 2008). Thus, even if the degree of control designed as input at the low levels of the Boulding's typology is illusionary, it can still lead to quite effective learning (Dror 2008).

In this case, control is perceived by the learners as the adaptivity of the system, which is responding to their choices, although the system actually provides only the illusion of control. At this point the hypothesis is restated as: "*If motivation is triggered through illusionary sense of control that resides in the collaborators, then a higher value of the QoC will be produced*", measured on the basis of the three aforementioned indicators.

The evaluation of the indicators is performed on the basis of elaboration and analysis of the logged raw data. When it is performed by an external to the collaborators agent, an in depth qualitative analysis of these data can be performed by experts; yet it would be a severe workload for them, as opposed to quantitative analysis performed by the system.

In this chapter, a combination of these approaches is presented towards the analysis of the collaborative interactions. More specifically, the use of the intuitionistic fuzzy-logic based approach is proposed, which provides a hybrid combination of the expert knowledge with the quantitative formalization. Moreover, it manages to capture the hesitation of the expert while evaluating, providing a more realistic and enhanced modeling approach as presented below.

18.5 FL/IFL-Based Quantitative Evaluation of QoC

18.5.1 The Fuzzy Logic Concept

System modelling requires the knowledge of its structure and function, and the existence of appropriate mathematical tools. When the system is complicated or ill-defined, conventional functions cannot be used to model it. But a domain expert, using natural language, could provide the required knowledge. In such cases, Fuzzy Logic (FL) technology, which was introduced by Zadeh (Zadeh 1965), allows modelling of the system using the expert's *linguistic descriptions* of it, in the form of If/Then rules (Tsoukalas and Uhrig 1996). Moreover, it allows the representation of these rules through *analytical forms* at the mathematical level.

More specifically, the domain expert linguistically describes his/her evaluating system of collaboration, using a series of If/Then rules. The syntax of these rules is facilitated by the linguistic variables. The qualitative characterization of the *variables*, namely *fuzzy values*, (e.g., *low*, *medium*, *high*), the connectives (e.g., *and, or, else*), the If/Then implication divides the rule into two parts, the hypothetical conditions, or antecedent and the inference, or consequent, which is true when the antecedent is satisfied.

According to the FL approach, a fuzzy value refers to a set of objects or meanings, which share common features. According to this notion, all the percentage values of the universe of discourse belong to a fuzzy value, yet to a *degree*. The degree ranges from 0 to 1, where 0 denotes zero membership of the set, and 1 denotes total membership of the set. This approach allows for infinite degrees of membership between 0 and 1, resulting in a more flexible representation of the word 'medium' from the linguistic to the analytical form.

This representation is analytically described by a *membership function*. Its shape is chosen subjectively, varying according to the expert's definition of the relevant set, yet on the basis of application specific criteria (Tsoukalas and Uhrig 1996). Thus, a membership function is a mechanism for mapping every *crisp value* of the universe of discourse of *initiative* to the interval [0, 1].

The connectives possess relevant *operators* at the analytical form, which conduct mathematical operations (e.g., *union* and *intersection*) (Tsoukalas and Uhrig 1996), realizing a *Fuzzy Inference System* (FIS) that interacts with the knowledge-base and performs the functions of the operators that are defined for the connectives, (i.e., *and/or, else* and the *If/Then* implication).

The latter possesses, at the analytical form, an *implication operator*, which combines the two parts of the rule to produce an *implication relation* (Tsoukalas and Uhrig 1996). When the antecedent of a rule is satisfied, the rule is activated (fired) and then through the *min If/Then implication operator*, the fuzzy number of the antecedent of the rule performs the clipping of an area from the membership function of the consequent part of the rule (Tsoukalas and Uhrig 1996).

Thus, the output of the *implication relation* is the remaining area of the membership function of the consequent part of the rule. In this way, a membership function area is produced for each rule of the knowledge-base that is fired. All these rules are then connected with the *else* operator to constitute the fuzzy algorithm that performs the inference procedure.

The *max* operator that corresponds to the connective *else*, aggregates all the clipped membership function areas and a synthesis of them is produced. This new area is the output of the evaluative inference (Tsoukalas and Uhrig 1996).

However, humans cannot interpret this information due to its fuzzy form; hence, a defuzzification process must occur. One common example is the centroid defuzzification, which returns the center of the output area.

18.5.2 Intuitionistic Fuzzy Logic

Description of system behavior in the language of fuzzy rules lowers the need for precision in data gathering and data manipulation, and in effect may be viewed as a form of data compression. But, there are situations when description by a fuzzy linguistic variable, given in terms of a membership function only, seems too rough. For example, in decision making problems, particularly in a case of medical diagnosis, sales analysis, new product marketing, financial services, etc., there is a fair chance of the existence of a non-null hesitation part at each moment of evaluation of any unknown object. Intuitionistic Fuzzy Logic (IFL) (Atanassov 1986, Atanassov 1999) can be viewed in the context as a proper tool for representing hesitancy concerning both membership and non-membership of an element to a set. The conclusion of an Intuitionistic Fuzzy Inference System (IFIS) can be defined as a linear combination of the results of two classical fuzzy systems, one corresponding to μ and the other to *v*, respectively, as given in equation (18.1)

$$IFIS = (1 - \mu)\ FIS\mu + \mu FISv \tag{18.1}$$

Where $FIS\mu$ is the traditional output of a FL system using the membership function μ and $FIS\mu$ is the output of a FL system using the non-membership function v, with $\pi > 0$. The advantage of this method for computing the output IFIS is that the previous machinery of traditional FL systems for computing $FIS\mu$ and $FIS\mu$ can be used, employing then just a weighted average of both results to obtain the final output IFIS (Castillo and Melin 2003). More on FIS and IFIS can be found in: Tsoukalas and Uhrig 1996, Atanassov 1986, Atanassov 1999. As stated in Sect. 18.3, the focus of the analysis of collaborative interactions is put upon the established indicators of *balance* (B), *productivity* (P) and *cognitive rate* (CR). These indicators are the outputs of three corresponding FIS that use the *turn-taking* (TT), the *N action types* (AT) and the *M entity types* (ET) logged by the CSCL setting as inputs, respectively. The estimated set of $\{B, P, CR\}$ is then fed into an IFIS that outputs the estimated *QoC* in an quantitative manner using Equation 18.1. Fig. 18.2 illustrates the general structure of the FIS/IFIS-based quantification of *QoC*.

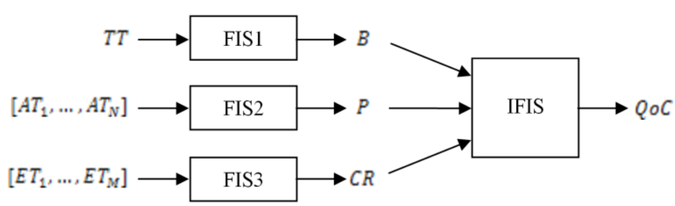


Fig. 18.2 The general structure of the FIS/IFIS-based quantification of QoC, TT, AT and ET denote turn-taking, action types (N in number), and entity types (M in number). E, P, and CR stand for balance, productivity and cognitive rate, respectively

The structure described in Fig. 18.2 could be implemented as a function of time, dynamically producing the *QoC* values per time unit. In our case, the $QoC(t)$ with $t = 1, 2\ldots L$ minutes is adopted, with L denoting the time duration of the whole collaborative session. This time resolution applied to all parameters of Fig. 18.2 seems practically reasonable, as it could establish both a natural collaboration flow between peers and a detailed monitoring of the evolution of the quality of peers' collaboration during their collaborative session.

The realization of FIS/IFIS depends upon the knowledge base adopted. The latter contains the expert's evaluating system, in terms of the analytical form of If/Then rules and membership functions, as they were previously described. As is evident, the construction of the knowledge base is a prerequisite for the inference procedure. It holds the knowledge upon which the evaluation takes place. Its content may vary, reflecting the relevant variations in the evaluation systems of different experts. This fact, however, does not diminish the value of the proposed approach, which focuses on the enhanced characteristics of the FL and IFL technology for the development of the expert evaluating system of interest.

In addition, the format of the acquired input (i.e., $\{TT, AT, ET\}$) depends upon the settings of the CSCL environment. In any case, an ordinary numerical coding could be adopted (e.g., -1 for Peer-*A* and +1 for Peer-*B*; hence the zero crossing could provide the *TT* value) to encounter the occurrence of each input parameter in the CSCL system log files. Sect. 18.6, presents an implementation of the whole approach described so far.

18.6 Experimental Case Study: Concept Mapping

The case study was designed upon a CSCL setting that referred to the construction of a concept map during the collaboration of two peers. Concept maps are graphical tools for organizing and representing knowledge. They include concepts, usually enclosed in circles or boxes of some type, and relationships between concepts indicated by a connecting line linking two concepts. Words on the line, referred to as linking words or linking phrases, specify the relationship between the two concepts. The latter are defined as *a perceived regularity in events or objects, or records of events or objects, designated by a label* (Novak and Cañas 2008).

Concept maps were set in 1972 in the course of Novak's research program at Cornell where he pursued to follow and understand changes in children's knowledge of science (Novak and Musonda 1991). His program accounted the learning psychology of Ausubel (Ausubel 1963, Ausubel 1968, Ausubel et al. 1978).

The key idea in Ausubel's cognitive psychology is that learning takes place by the *assimilation* of new concepts and propositions into existing concept and propositional frameworks held by the learner. This knowledge structure as held by a learner is referred to as the individual's *cognitive structure*. The idea of depicting children's knowledge in a concept map emerged out of the necessity to find a better way to represent children's conceptual understanding (Novak and Cañas 2008).

One of the reasons concept mapping is so powerful for the facilitation of meaningful learning is that it serves as a kind of *template* or *scaffold* to help to organize knowledge and to structure it, even though the structure must be built up piece by piece with small units of interacting concept and propositional frameworks. The IHMC CmapTools (Cañas et al. 2004) (available at http://cmap.ihmc.us) provides an implementation of the concept mapping in a Web-based environment, allowing the user to link resources (photos, graphs, videos, charts, tables, texts, html pages or other concept maps) located anywhere on the Internet or in personal files to concepts or linking words in a concept map through a simple drag-and-drop operation. The CmapTools supports synchronous, asynchronous, and distant collaboration and provides all necessary log files that relate with peers' software interaction.

18.6.1 Design and Implementation Issues

18.6.1.1 The Participants

Two experimental uses of the IHMC CmapTools version 5.3 were designed. The control and the experimental conditions were called Exp-A and Exp-B, respectively. The participants, who collaborated in 11 groups of dyads (namely $G_{i,\,f}$ = 1, 2…11) of mixed gender in both experiments, were undergraduate education students, aged between 19-20 years old.

They were completely acquainted with the collaborative concept mapping using the IHMC CmapTools about a month before the experiments. A 15 (±1) min duration (*L* value) of the collaborative session was considered adequate enough to keep the procedure intensive, based upon the findings of previous studies (Hadjileontiadou et al. 2010, Hadjileontiadou et al. 2011).

18.6.1.2 The Setting

During the Exp-A, the students collaborated without any perception of being monitored by the system regarding their collaborative activity. The theme of the concept map was left to the choice of the peers and was decided before the beginning of the Exp-A. Two weeks later (reset time period), the Exp-B, was conducted within the same groups and the same concept map theme per group, but they were told that the system continuously monitors their collaboration and adaptively estimates their *QoC* according to their collaborative interactions.

The IHMC CmapTools did not take into account such adaptivity possibilities. However, the information provided to the participants aimed at causing intrinsic motivation towards better collaboration through the cultivation of the idea that they could control the assessment of their collaborative performance (illusionary sense of control).

The peers had no knowledge as to whether the system actually estimated their *QoC*; they just had to accept what was told to them. In addition, no rewarding feedback was provided to them during the session, letting them control their collaborative activity according to their individual sense of *QoC*.

The choice of sustaining the same groups and the same concept map theme was decided on to make the monitoring of the effect of the illusion of control to the peers independent of the group composition and task content. This within-subject analysis allowed for higher statistical power, as participants were compared to themselves, problems of group differences in between-subject designs were avoided, and it was easier to find significant difference in the dependent variables, due to the reduction in variability.

In order to reinforce the illusion of monitoring the peers' collaborative activity by the system, a text-feedback in a reminder-like form ("Remember: the system keeps records of each peer's collaborative activity") was provided to all groups around the middle of the collaborative session. This reminder was kept as neutral as possible, in order to keep unaffected the peers' collaboration and individual interpretation of *QoC*, focusing only on the sustainment of their illusion of control.

18.6.1.3 Data Collection and Elaboration

The interactions in both experimental conditions were recorded in IHMC CmapTools and log files were exported in .txt format. For each interaction, the log files included the number, time stamp, user's identity (-1 for Peer-*A* and +1 for Peer-*B*), action type (At = [Add, Delete, Modify Text]; $N = 3$) and entity type (Et = [Concept, Linking Phrase, Connection]; $M = 3$). The quantitative elaboration of these raw data was followed by processing and statistical analysis using Matlab R2009b (The Mathworks, Inc., Natick USA). Values of $p < 0.05$ were considered as indicators for statistically significant differences.

18.6.1.4 FIS/IFIS Realization

All FIS and IFIS were of Mamdani-type inference systems (Tsoukalas and Uhrig 1996), adopting the 'and' to 'min', 'or' to 'max', 'If/Then implication' to 'min', and 'else aggregation' to 'max' correspondences. The defuzzification method used was the *'centre-of-mass'* (Hines 1997).

The membership functions both for inputs and outputs (both for FIS and IFIS) were of trapezoid shape, while the fuzzy values per input in FIS were {*very low*, *low*, *medium*, *high, very high*} for the *TT* input and {*low*, *medium*, *high*} for the *AT* and *ET* inputs.

For example, the values of the *ADD* variable of the *AT* FIS lying within the interval of 0-30 were corresponded to {*low*: 0-12; *medium*: 6-24; *high*: 18-30}. In all cases the fuzzy values for all FIS outputs were of {*low*, *medium*, *high*}.

For the case of IFIS, for all inputs the fuzzy values of {*low*, *medium*, *high*} were used, whereas for the IFIS output the fuzzy values of {*low*, *medium*, *high, very high*} were adopted. The π value in Equation 18.1 was set to 0.075 and $\pi_A(x)$ was applied to both sides of the fuzzy value "Medium" of all IFIS inputs and output, as there the expert's hesitancy is maximized. The rule base was built by an expert, involving all possible combinations of the inputs (i.e., max 27 rules for each FIS/IFIS) to the formulation of the corresponding output. All FIS/IFIS outputs lied within [0, 1]. The '*.fis' MATLAB files are available upon request.

18.6.2 Results

In order to accept or reject the hypothesis, the values of the inputs to the FIS sketched in Fig. 18.2 were estimated, as they are presented in Fig. 18.3. More specifically, Fig. 18.3(a) depicts the number of *TT* in both experimental conditions (Exp-A and Exp-B), for the total experiment duration of each $G_{f,\mathrm{i}} = 1, 2 \ldots 11$ Figs. 18.3(b) and 18.3(c) depicts the total number of actions and entities across all groups, per action and entity type, respectively, for both experimental conditions.

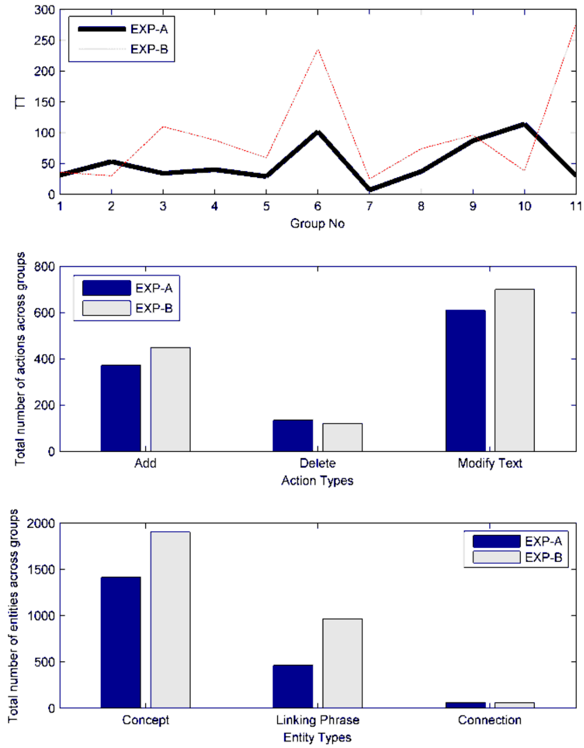
In Fig. 18.3, it is evident that a general increase in the number of the collaborative interactions is produced under the Exp-B. The cumulative results presented in Fig. 18.3 are expanded to the dynamic ones, by employing the time-based analysis, i.e., estimating $TT(t)$, $AT_i(t)$, $ET_i(t)$, $i = 1, 2, 3$; $t = 1, 2 \ldots L$ (min), and calculating the corresponding FIS outputs of $B(t)$, $P(t)$ and $CR(t)$, $t = 1, 2 \ldots L$ (see Fig. 18.2); the latter are illustrated in Fig. 18.4 to 18.6, respectively. There, the time axis was normalized to the L_i value of each group ($i = 1, 2 \ldots 11$), so a unified scale across groups of the experiment duration in % is established.

The estimated dynamic outputs from FIS drawn in Fig. 18.4 to 18.6 were subsequently fed to the IFIS (see Fig. 18.2), resulting in the $QoC(t)$ set in Fig. 18.7.

18.6.3 Discussion

Looking at the presented results, both from the perspective of the initial inputs, the intermediate and final outputs, the acceptance of hypothesis (i.e., that the provided illusionary sense of control could impact on the *QoC*, can be supported).

More specifically, from Fig. 18.3(a) it can be noted that the *TT* curve (transition of interactions between the two peers) of Exp-B exceeds the corresponding *TT* curve of Exp-A, showing in 8 out of the 11 groups, i.e., G_1, G_2: G_g, and G_{11}, a clear increase, with the higher differences located at G_g, and G_{11}. This finding indicates that the collaboration flow, in terms of the number of the *TT* of the interactions performed at the experimental condition Exp-B, was higher than that in the control one (Exp-A) ($p < 0.05$; Wilcoxon Ranksum test within subjects).



(a)

(b)

(c)

Fig. 18.3 (a) Total number of TT during *the* experiment duration per group, and experiment. (b) Total number of actions across all groups, per action type and experiment. (c) Total number of entities across all groups, per entity type and experiment

Moreover, from Fig. 18.3(b) it can be noticed that in Exp-B the total number of actions, in all cases but the case of action type AT_2^{EXP-B} = Delete, increases ($p <$ 0.05; Wilcoxon Ranksum test within subjects), indicating a positive effect of the illusion.

The decrease in AT_2^{EXP-B} could be possibly explained by the increase in AT_3^{EXP-B} = Modify Text, since it seems that the peers, under the illusion effect, preferred the modification process rather than the deletion one. This provides a kind of sustainment in the collaboration, as the deletion process is almost instant, whereas the modification process requires more time, triggers negotiation processes and refinement actions that contribute towards higher peers' engagement in the collaborative activity.

The latter is further augmented with the increase in the AT_1^{EXP-B}=Add ($p < 0.05$; Wilcoxon Ranksum test within subjects), showing an overall productive collaborative attitude under the existence of illusion, since both Exp-A and Exp-B refer to the same conditions (e.g., duration, concept map theme, pairing), differing only in the provision of illusionary control.

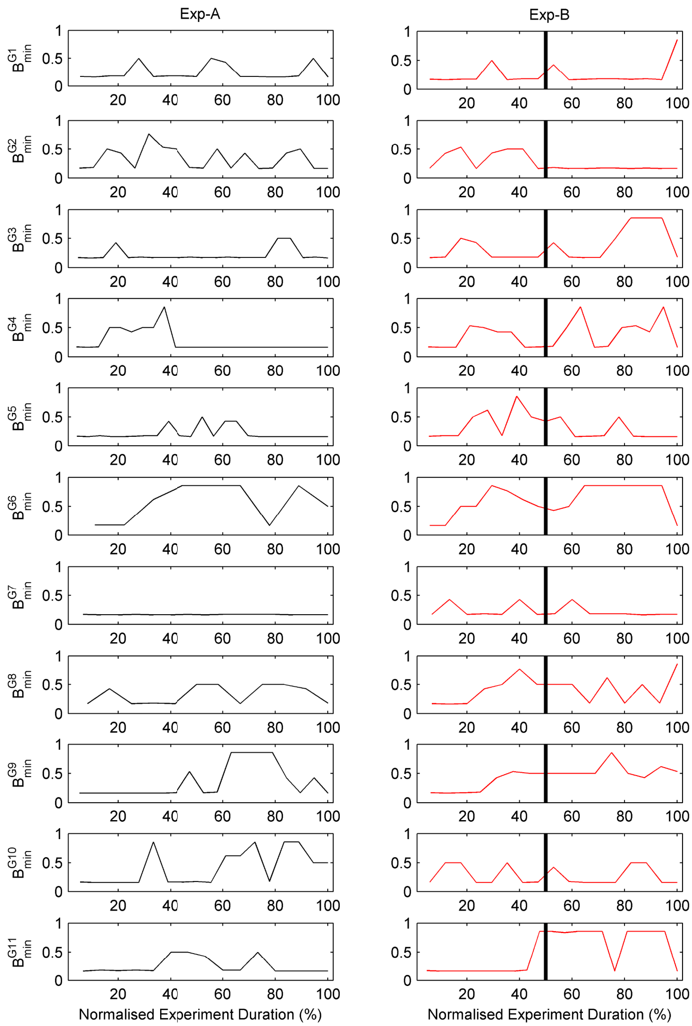


Fig. 18.4 The estimated balance per min $B_{\min}^{G_i}(t), t = 0{:}\,100\%\ L_i, i = 1,2,\ldots,11$, per group ($G_i$), and experimental condition (Exp-A, Exp-B). The vertical line in Exp-B denotes the occurrence of the control reminder

A similar finding is deduced by the results presented in Fig. 18.3(c), where there is a distinct increase in two out of three ET, i.e., ET_1^{EXP-B} = Concept and ET_2^{EXP-B}=Linking Phrase, for the case of Exp-B compared to Exp-A ($p < 0.05$; Wilcoxon Ranksum test within subjects); there is no change in the ET_3 =Connection. The increase in ET_1^{EXP-B}justifies the fundamental idea in Ausubel's cognitive psychology regarding the construction of the individual's cognitive structure by the assimilation of new concepts into existing ones.

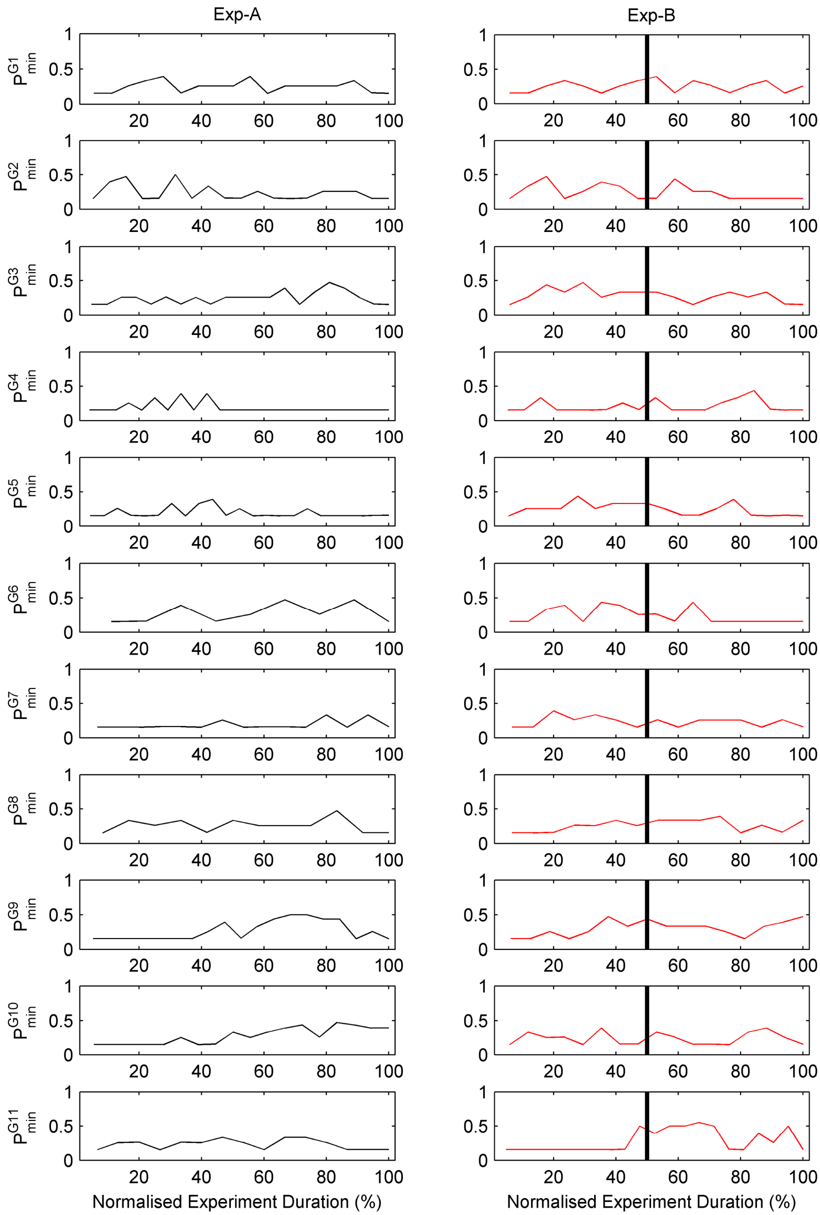


Fig. 18.5 The estimated productivity per min $P^{G_i}_{\mathrm{min}}(t), t = 0{:}\,100\%\,L_i, i = 1,2,\ldots,11,$ per group (G_i,) and experimental condition (Exp-A, Exp-B). The vertical line in Exp-B denotes the occurrence of the control reminder

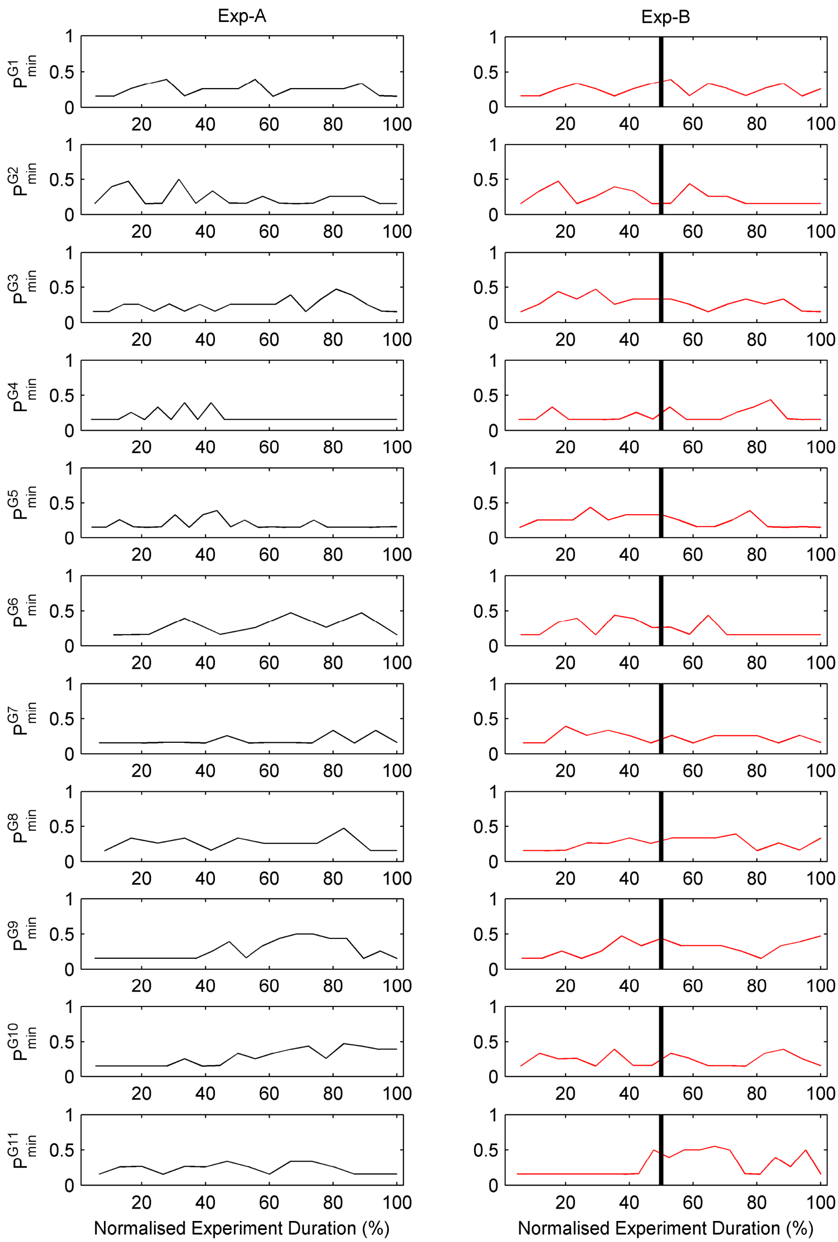


Fig. 18.6 The estimated cognitive rate per min $CR^{G_i}_{\min}(t), t = 0\!:\!100\%\ L_i, i = 1,2,\ldots,11$, per group ($G_i$), and experimental condition (Exp-A, Exp-B). The vertical line in Exp-B denotes the occurrence of the control reminder

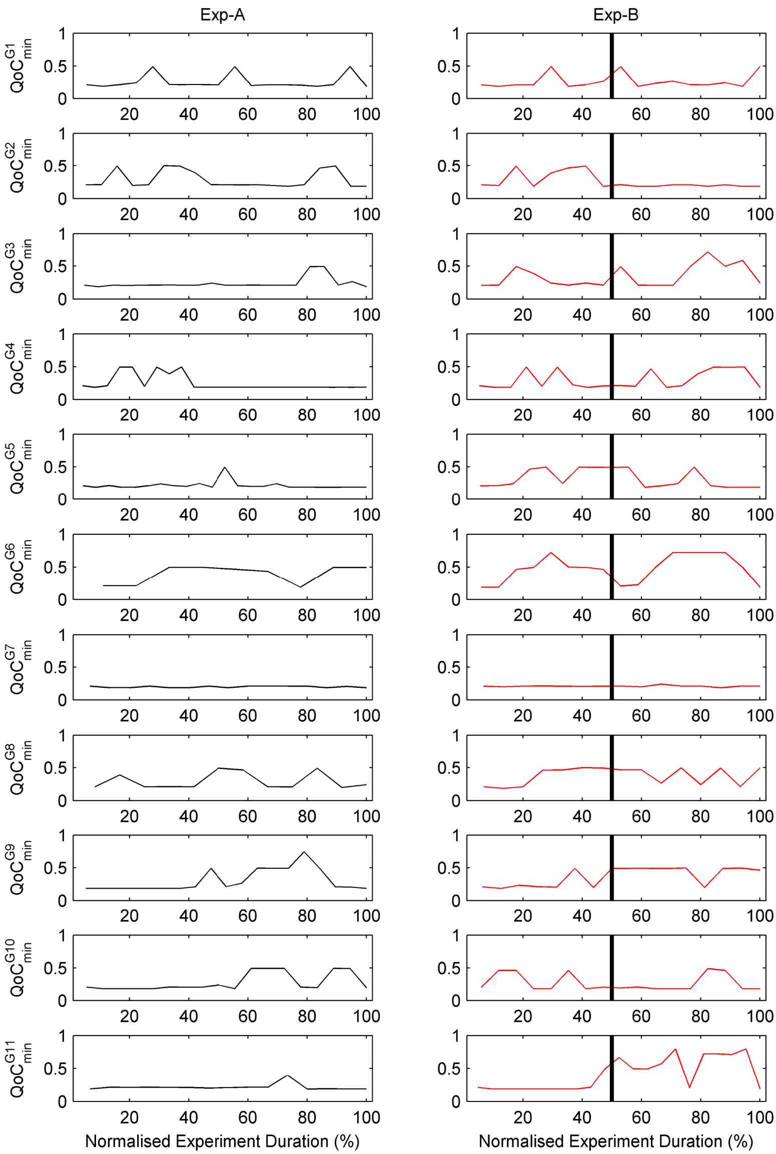


Fig. 18.7 The estimated quality of collaboration per min $QoC^{G_i}_{\min}(t), t = 0{:}\,100\%\, L_i,\ i = 1,2,\ldots,11$, per group ($G_i$), and experimental condition (Exp-A, Exp-B). The vertical line in Exp-B denotes the occurrence of the control reminder

Consequently, the existence of illusion contributes to the enrichment of peers' cognitive structure expressed in their interactions within the CSCL setting. Furthermore, the increment in ET_2^{EXP-B} reveals increase in peers' relational activity through the introduction of relationships or links between concepts in different segments or domains of the concept map. This is also important in the creation of new knowledge, as cross-links often represent creative leaps on the part of the knowledge producer. Consequently, the illusionary sense of control shows a creative contribution to the peers' total collaborative activity, in terms of flow, activity and cognitive structure.

Moving onto the more abstract level of representation of the collaborative activity, shifting to the *balance*, *productivity* and *cognitive rate* parameters, the results presented in Figs. 18.4-18.6, respectively, reveal the dynamic character of peers' collaboration.

In particular, looking at Fig. 18.4, a more balanced behavior, in terms of equal contribution of both peers in the collaborative activity, is noticed for the case of Exp-B compared to the Exp-A one, in all groups except G_2. For example, in the case of G_1, the $B_{min}^{G_4,EXP-A}$ shows an almost constant behavior across the whole duration of Exp-A, with an exception around the 15-40% L_1, whereas this unbalanced behavior is rectified with the existence of illusion, since $B_{min}^{G_4,EXP-B}$ exhibits higher values, almost for the whole duration of Exp-B.

A similar case can be seen for the G_1, where the constant unbalanced collaboration during Exp-A is improved towards a more balanced one during Exp-B, at least for 50% of its duration. The contribution of the illusion reminder at the 50% of the experiment duration is quite obvious in the case of G_{11}, where there is a distinct change in the peers' balance after the 50% of the L_{11}, shifting $B_{min}^{G_{11},EXP-B}$ from a constant low value (<0.2) to a very high one (>0.95). This is also noticed across almost all groups (except $G_{2,\ 5,\ 7}$), as the cumulative sum of the balance activity post-reminder is higher than the one pre-reminder.

In order to elaborate upon the role of the reminder serving as an important factor in Exp-B, a few other comparison points were examined to knock down the competing hypothesis of practice affecting the performance of the skill more than the illusionary sense of control. In this vein, the cumulative sum of balance activity from the intervals 0-25%, 25-50%, 50-75% and 75-100% of the duration L_1 in the form of box plot across groups was estimated. The results appear in Fig. 18.8.

From Fig. 18.8 it is clear that the median value (horizontal line within each box) of the cumulative sum of the balance activity across the intervals does not show any continuous rising trend; a fact that, if true, would support the "practice time" hypothesis. Moreover, in Fig. 18.8, the inter-quartile range (box height) tends to increase across the intervals, increasing simultaneously the deviation from any underlying trend. Consequently, the "practice time" hypothesis could, in general, be rejected, justifying the effective role of the reminder.

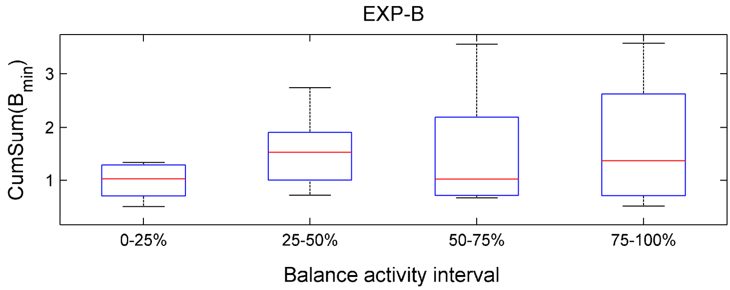


Fig. 18.8 Boxplot of the cumulative sum of balance activity from the intervals 0-25%, 25-50%, 50-75% and 75-100% of the duration L_1 across all groups. Horizontal line in the box indicates the median value; the box height denotes the inter-quartile range between the first and the fourth quartile of each distribution; the whiskers show 1.5 times the standard deviation

Similar behavior as in the case of *balance* can be seen in the cases of *productivity* and *cognitive rate* (Figs. 18.5 and 18.6 and the corresponding box plotting, like Fig. 18.8, not shown here), where the existence of illusionary sense of control fosters the productivity and the cognitive rate to higher values, when compared to the ones from the Exp-A, for longer intervals during the collaborative sessions.

When moving to the final output of the FIS/IFIS structure of Fig. 18.2, i.e., $QoC(t)$, presented in Fig. 18.7, the hypothesis is clearly justified, since indeed the existence of the illusionary sense of control in the Exp-B has contributed to higher values of $QoC(t)$ (stronger difference with $p < 0.01$; Wilcoxon Ranksum test within subjects), compared with the ones in Exp-A.

As in the previous perspective, the illusion reminder has positively affected most of the groups (e.g., G_3:G_6. G_8, G_9, G_{11}) to sustain and/or increase their $QoC(t)$. This finding shows a *common strategy*-like of collaboration that was adopted by almost all groups, where the collaboration between peers is built during the collaborative session, starting with different delays in the establishment of adequate QoC towards a 'maturing' process of collaboration. The results of Fig. 18.7 indicate, in many cases (see for example G_2, G_5, G_9 : G_{11}), the existence of illusion reduced these delays in the maturing process of peers' collaboration.

The above empirical results allow the acceptance of the hypothesis and extend the findings of Vandewaetere et al. 2009, about the enhancement of the learning outcome through the perceived illusion of control, from individual to collaborative settings. This effort was grounded on the conceptual framework of the Boulding's typology and contributed to its further elaboration in the area of CSCL. This framework reveals dualities within the complex social system of the collaborating dyad (e.g., designable-undesignable inputs, stable-adjustable levels, external input-internal attractors). These clarifications allow the realization of the form of the illusion (information), the timing of its provision (diminishing the assistance dilemma), the understanding of its perception mechanisms, and the expected impact at the social plane.

The approach described in this chapter allows handling of the intangible notion of illusion, at no cost and computational effort, within a CSCL setting that did not take into account any adaptive character. Thus, it moved the issue of adaptation of the support provided to the collaborators from 'hard' system modeling designs to more 'soft' ones, revealing possibilities of further elaborations of the boundaries and contents of the aforementioned levels.

18.7 Conclusions

In this chapter, an IFL-based approach of intrinsic motivation in CSCL settings during illusionary sense of control was proposed that elaborates the Boulding's typology in the area of CSCL. Within the hierarchy of this typology, the focus was put on the higher levels, where the perception of adaptivity in the form of control can be illusionary triggered, in order to increase the possibilities to sustain the even higher level of social collaborative work. The way this is internalized by the peers and expressed through their collaborative interaction within the CSCL setting was modeled by the structure of an IFIS, which, combined with classic FIS, has resulted in a dynamic quantitative evaluation of the quality of collaboration across the peers' collaborative activity. This approach was validated through an experimental study that involved joint construction of concept maps, verifying that the illusion of the adaptivity in the form of input provided at the lower levels of this typology, may enhance peers' collaborative performance.

The main concepts and the promising experimental findings presented in this chapter need to be justified through large-scale experiments. Nevertheless, in any case, they shed light upon the process of expression of the collaborative interactions and offer better understanding of their underlying mechanisms, by extending the capabilities of tracking their 'fine-grained' processes. Finally, it is our hope that the issues discussed in the present chapter will provide an alternative framework for structuring efficient CSCL settings, taking into account the relation between perception of adaptivity and learning outcomes.

Acknowledgments. The authors would like to thank the 22 participants in the experimental case-study for their willingness to support such research endeavor. Moreover, the authors acknowledge the contribution of Professor Elizabeth Owen Bratt to the hypothesis testing process.

References

Abramson, J., Mizrahi, T.: When social workers and physicians collaborate: Positive and negative interdisciplinary experiences. Social Work 41(3), 270–281 (1996)

Atanassov, K.: Intuitionistic fuzzy sets. Fuzzy Sets and Systems 20(1), 87–96 (1986)

Atanassov, K.: Intuitionistic fuzzy sets: Theory and applications. Physica-Verlang, Heidelberg (1999)

Ausubel, D.P.: The psychology of meaningful verbal learning. Grune and Stratton, New York (1963)

Ausubel, D.P.: Educational psychology: A cognitive view. Holt, Rinehart and Winston, New York (1968)

Ausubel, D.P., Novak, J.D., Hanesian, H.: Educational psychology: A cognitive view, 2nd edn. Holt, Rinehart and Winston, New York (1978)

Berg-Weger, M., Schneider, F.: Interdisciplinary collaboration in social work education. Journal of Social Work Education 34(1), 97–107 (1998)

Boulding, K.: General systems theory-The skeleton of science. Management Science 11(3), 197–208 (1956)

Cañas, A., Hill, G., Carff, R., Suri, N., Lott, J., Eskridge, T.: CmapTools: A knowledge modeling and sharing environment. In: Cañas, A., Novak, J., González, F. (eds.) Proceedings of ICCM, pp. 125–133. Universidad Pública de Navarra, Pamplona (2004)

Castillo, O., Melin, P.: A new method for fuzzy inference in intuitionistic fuzzy systems. In: Proceedings of NAFIPS, pp. 20–25. IEEE Press, New York (2003)

Checkland, P.B.: Systems thinking, systems practice. Wiley, New York (1981)

Corbalan, G., Kester, L., Merriënboer, J.: Combining shared control with variability over surface features: Effects on transfer test performance and task involvement. Computers in Human Behavior 25(2), 290–298 (2009)

Deci, E., Ryan, R.: The “what” and "why" of goal pursuits: Human needs and the self-determination of behavior. Psychological Inquiry 11(4), 227–268 (2000)

Dillenbourg, P.: Collaborative learning cognitive and computational approaches. Pergamon, Oxford (1999)

Dillenbourg, P., Baker, M., Blaye, A., O’ Malley, C.: The evolution of research on collaborative learning. In: Spada, E., Reiman, P. (eds.) Learning in Humans and Machines: Towards an Interdisciplinary Learning Science, pp. 189–211. Elsevier, Oxford (1995)

Dror, I.: Technology enhanced learning: The good, the bad, and the ugly. Pragmatics & Cognition 16(2), 215–223 (2008)

Elen, J., Clarebout, G.: The use of instructional interventions: Lean learning environments as a solution for a design problem. In: Elen, J., Clark, R. (eds.) Handling Complexity in Learning Environments: Research and Theory, pp. 185–200. Elsevier, Amsterdam (2006)

Gabriele, S.: Boulding’s typology elaborated: A framework for understanding school and classroom systems. Systems Practice 10(3), 271–303 (1997)

Gabriele, S.: The hard facts of soft social systems: Towards a theoretical and practical model for schools and other organizations. International Society for the Systems Sciences (2010), `http://journals.isss.org/index.php/proceedings54th/search/authors?searchInitial=G` (accessed May 25, 2011)

Green, R., Etheridge, C.: Building collaborative relationships instruction improvement. Education 120(2), 388–397 (1999)

Hadjileontiadou, S., Sakonidis, H., Balafoutas, G.: Link2: a novel web-based collaborative tool-application to engineering education. Journal of Engineering Education 92(4), 313–324 (2003)

Hadjileontiadou, S., Nikolaidou, G., Hadjileontiadis, L.J.: Illusionary adaptive support as a means for intrinsic motivation in CSCL settings: A case in concept mapping. In: Jimoyiannis, A. (ed.) Proceedings of PCI, pp. 163–170. University of Peloponnese, Korinthos (2010)

Hadjileontiadou, S., Nikolaidou, G., Hadjileontiadis, L.J.: Instructional design on controlling the quality of collaboration in a CSCL setting through illusionary adaptive support. In: Jimoyiannis, A. (ed.) Research on E-Learning and ICT in Education, pp. 141–154. Springer Science+Business Media, New York (2011) (in press)

Harter, S., Connell, J.: A model of children's achievement and related self-perceptions of competence, control, and motivational orientation. In: Nicholls, J.G. (ed.) Advances in Motivation and Achievement: The Development of Achievement Motivation, pp. 219–250. JAI Press, Greenwich (1984)

Hines, W.: Fuzzy and neural approaches in engineering. Wiley, New York (1997)

Holmes, G., Howson, C.: Grants: interdepartmental collaboration to teach grantsmanship skills. The Bottom Line: Managing Library Finances 13(3), 146–149 (2000)

Kapur, M., Rummel, N.: The assistance dilemma in CSCL. In: Dimitrakopoulou, A., O' Malley, C., Suthers, D., Reimann, R. (eds.) Proceedings of CSCL, vol. 2, pp. 37–39. International Society of the Learning Sciences, New Jersey (2009)

Katz, I., Assor, A.: Is autonomy important for non-western students? Examining autonomy as a universal human propensity. In: Proceedings of AERA (2003)

Katz, I., Assor, A.: When choice motivates and when it does not. Educational Psychology Review 19(4), 429–442 (2007)

Kinzie, M.: Requirements and benefits of effective interactive instruction: Learner control, self-regulation, and continuing motivation. Educational Technology, Research and Development 38(1), 5–21 (1990)

Kinzie, M., Sullivan, H.: Continuing motivation, learner control and CAI. Educational Technology Research and Development 37(2), 5–14 (1989)

Meier, A., Spada, H., Rummel, N.: A rating scheme for assessing the quality of computer-supported collaboration processes. International Journal of Computer-Supported Collaborative Learning 2(1), 63–86 (2007)

Novak, J., Cañas, A.: The theory underlying concept maps and how to construct and use them. IHMC CmapTools (2008), `http://cmap.ihmc.us/publications/researchpapers/theorycmaps.htm` (accessed March 1, 2011)

Novak, J., Musonda, D.: A twelve-year longitudinal study of science concept learning. American Educational Research Journal 28(1), 117–153 (1991)

Paas, F., Tuovinen, J., Merrilnboer, J., Darabi, A.: A motivational perspective on the relation between mental effort and performance: Optimizing learner involvement in instruction. Educational Technology Research and Development 53(3), 5–34 (2005)

Rogoff, B.: Observing sociocultural activity on three planes: participatory appropriation, guided participation, and apprenticeship. In: Goodnow, P.M.J., Kessel, F. (eds.) Sociocultural Studies of Mind, pp. 139–164. Cambridge University Press, Cambridge (1995)

Stahl, G., Koschmann, T., Suthers, D.: Computer-supported collaborative learning: An historical perspective. In: Sawyer, R.K. (ed.) Cambridge Handbook of the Learning Sciences, pp. 409–426. Cambridge University Press, Cambridge (2006)

Tsoukalas, L., Uhrig, R.: Fuzzy and neural approaches in engineering. Wiley, New York (1996)

Vandewaetere, M., Clarebout, G., Desmet, P.: The illusion of adaptivity as instructional method in electronic learning environments. International AIED Society (2009), `http://celstec.org/system/files/file/conferenceproceedings/aeid2009/papers/paper_129.pdf` (accessed May 25, 2011)

Walther-Thomas, C., Korinek, L., McLaughlin, V.: Collaboration to support student's success. Focus on Exceptional Children 32(3), 1–18 (1999)

Wertsch, J., Stone, C.: A social interactional analysis of learning disabilities remediation. Association for Children with learning Disabilities (1979), http://pdfserve.informaworld.com/54239_758077517_916540222.pdf (accessed May 25, 2011)

Williams, M.: Learner-control and instructional technologies. In: Jonassen, D.H. (ed.) Handbook of Research on Educational Communications and Technology, pp. 957–982. MacMillan Publishers, New York (1996)

Wulf, A.: The Collaboratory opportunity. Science 261(5123), 854–855 (1993)

Zadeh, L.: Fuzzy sets. Information and Control 8(3), 338–353 (1965)

Abbreviations

AT	Action Types
B	Balance
QoC	Quality of the Collaboration
CR	Cognitive Rate
CSCL	Computer Supported Collaborative Learning
ET	Entity Types
Exp-A	Experimental Condidtion-A
Exp-B	Experimental Condidtion-B
FIS	Fuzzy Inference System
FL	Fuzzy Logic
H	Hypothesis
IFIS	Intuitionistic Fuzzy Inference System
IFL	Intuitionistic Fuzzy Logic
P	Productivity
TT	Turn-Taking

Chapter 19
An Intelligent System for Modeling and Supporting Academic Educational Processes

Setsuo Tsuruta[1], Rainer Knauf[2], Shinichi Dohi[1], Takashi Kawabe[1], and Yoshitaka Sakurai[3]

[1] School of Information Environment, Tokyo Denki University, Inzai, Chiba, 270–1382, Japan
{ysakurai,dohi,tsuruta}@sie.dendai.ac.jp
[2] Faculty of Computer Science and Automation, University of Ilmenau
98684 Ilmenau, Germany
rainer.knauf@tu-ilmenau.de

Abstract. University has a complicated system of course offerings, registration rules, and prerequisite courses, which should be matched to students' dynamic learning needs, and desires. We address this problem by developing an Educational-Learning System called "Dynamic Storyboarding System". Besides modeling learning processes, this system aims at evaluating and refining university curricula to reach an optimum of learning success in terms of best possible ac-cumulative *grade point average* (GPA). This is performed by applying Educational Data Mining (EDM) to former students curricula and their degree of success (GPA) and thus, uncovering golden didactic knowledge for successful education. It consists of mining a decision tree (DT) and applying it to curricula planned by current students. Students receive an estimation of the GPA they are likely to receive along with a recommendation to supplement a partial path to reach optimal success. Our approach includes individual learner profiles. The profiling concept initially uses the per-university educational history and is dynamically extended by the students' university study results. The profiles are used by applying the EDM technology to students with profiles of a high similarity to the student under consideration. A feasibility study showed the usefulness of the system. The effect has been validated by cross-validation with about 200 students' records. The mean of the difference between the original grade point average (GPA) and the estimated one was 0.43 with a standard deviation of 0.30.

19.1 Introduction

Academic educational processes often suffer from a lack of didactic design. Since universities are also research institutions, their professors are usually employed based on their topical skills.

A. Peña-Ayala (Ed.): Intelligent and Adaptive ELS, SIST 17, pp. 469–496.
springerlink.com

Didactic skills are often underestimated in the recruiting process. We refrain from discussing reasons for that, but focus on the issue of providing didactic support for university teaching. The application of didactic skills in teaching situations is not formally modeled for use in academic education. Moreover, much of such skills are not represented at all, but just "implemented" in the heads of experienced instructors.

To make didactic design explicit, we developed a (semi) formal modeling approach called *storyboarding*. Besides providing didactic support, storyboarding is setting the stage to apply Knowledge Engineering Technologies to verify, validate, and refine the didactics behind a learning process. The verification may include formally checking both logical consistency and didactic issues.

Moreover, didactics can be refined according to revealed weaknesses and proven excellence based on students' storyboard paths and their related learning success in terms of their achieved GPA. Didactic patterns can be explored by applying EDM techniques to the various ways students went through a storyboard and their associated level of success in terms of GPA.

Applying Data Mining (DM) to social processes and an estimation of human behaviour or its results is a general trend in Information Technology (IT). For marketing or voting processes this is done by means of software such as: SPSS Modeler and STATISTICA Data Miner (Karl Rexer et al. 2009).

For human learning processes, this is still in the fledging stages, but it assures success, helped by didactic knowledge representation power of a storyboard introduced here. As a result, future instructors and students may utilize these results by choosing evidentially successful ways through a storyboard.

The technology is adaptive in terms of computing the mining results due to both the educational history of the considered student and the data base whose data is dynamically updated by the students' study results for the DM technology.

A storyboard provides a road map for a lesson, a course, a subject to teach, or a complete study. To fulfill different learning needs, it includes alternative paths and possible detours if certain concepts to be learned need reinforcement. Using modern media technology, a storyboard also plays the role of a server that provides the appropriate content materials as deemed necessary. The modular and hierarchic character of a storyboard supports the re-use of components (Episodes and Scenes) for other (new) storyboards. There are at least three dimensions in which our modeling approach differs from others: expressiveness, the degree of being domain-based and IT-based complexity (Knauf et al. 2010).

Our storyboard concept is built upon standard concepts and enjoys (1) clarity by providing a high-level modeling approach, (2) simplicity, which enables everybody to become a storyboard author, and (3) visual appearance as graphs.

Storyboards represent the didactics behind learning in general and provide appropriate paths in the system of nested graphs, depending on different students needs. Storyboarding in the present chapter is expected to contribute towards considering the learning in a wider context, not just a particular course or subject.

This is because it can give qualified assistance for students to keep an overview and respect regulations in the maze of opportunities and limitations on university education. Learning activities need to be composed and designed at different levels. A fine grained level is the design of a lesson or, even finer, the discussion of a particular problem.

The larger the scope of learning is, the more the human activities are involved in the management of comprehensive learning activities. In other words, by enlarging the scope, the subject modeled by storyboards is extended from only didactics to didactics and management. For example, shifting from a severe theory to a small example for its explanation is a tactic didactic decision that can be easily made by the teacher without worrying that much about resources or the cooperation of other individuals outside the current scene. Inserting an additional subject in a curriculum, on the other hand, may have larger impacts on resource issues like time consumption, costs, availability of a related offer for the desired semester and so on. Issues like these add management issues to the pure didactic issues of composing or modifying a fine grained storyboard.

Through adopting the storyboard concept for a complete university study, the management of the study also becomes accessible for evaluation and refinement, (i.e., quality assurance). As a deeper benefit of this work, an EDM technology can be performed over the paths of particular students after they have completed their study at the Tokyo Denki University (TDU). This work answers questions like:

- What do the successful students' paths have in common?
- What do their paths distinguish from the ones of less successful students?

Likewise, the research pursues to enable students to create curricula with optimal success in terms of GPA. The approach is based on EDM technology, which infers typical personalized "success patterns" derived from students with good GPA and utilizes them for a GPA estimation of curriculum suggestions of current students as well as their adjustment with respect to better success chances based on former successful students' paths.

Thus, our system, called Dynamic Learning Need Reflection (DLNR), holds and utilizes the "collective experience of former students" and serves as a benefit for current students. In particular, it estimates the accumulative GPA of a curriculum planned by a student in advance to performing this curriculum and gives advice of how to refine it to increase this estimated value. Before we show the way to adapt the storyboarding concept for the intended application, a short introduction to the DLNR system is introduced in the next section for better understanding.

19.2 The Dynamic Learning Need Reflection System

DLNRS primarily aims at promoting the students' motivation by creating or modifying their own class schedule per semester or graduation time lines by themselves (Dohi & Nakamura, 2003).

This is a way to develop a spirit of independence and to keep up with globalization. Key features of DLNRS are as follows:

- Abolition of the traditional rigid academic year: There is no academic year with fixed courses and fixed fees. Instead, there is a semester-based course system with a tuition fee for each particular subject. There is no restriction for taking a particular subject in a particular semester except the prerequisites specified for the subject. Thus, the students are able to study at their own pace,
- Abolition of compulsory subjects: Specific compulsory subjects have been replaced by the concept of prerequisite conditions. These conditions are expressed in two levels of recommendation: subjects that have to be learnt before and subjects that are recommended to be learnt before. The prerequisites are formally checkable by considering the GP marked in the subjects that are prerequisites,
- Replacement of a fixed charge per year by a course-oriented paying system: Students pay a subject-oriented fee in proportion to the number of units of one subject. Therefore, they carefully check their learning needs to pick out the right subjects to achieve their academic goal. Furthermore, it motivates them to make a maximum effort to pass the subject to get their money's worth,
- Class period length: The usual length of a class is reduced from 90 minutes to 50 minutes or 75 minutes. Typically, a subject is taught in 3 units either as 3 *x* 50 minutes or 2 *x* 75 minutes a week. The intended effect is that students will be able to concentrate on the entire length of a class. Therefore, it is a contribution towards more benefits of learning gained from the subjects and thus, from the money spent for it,
- GPA: This is a system to rate the learning results and to derive appropriate consequences for the upcoming educational process schedule. The GPA of subjects is calculated by equation (19.1):

$$GPA = \frac{\sum_{i=1}^{n} u_i g_i}{\sum_{i=1}^{n} u_i} \tag{19.1}$$

Where g_i being the points earned for a particular subject, u is the number of units of the subject, and n corresponds to the number of subjects in the semester. According to the given percentages of performance GP is instantiated as follows:

$$GP = \begin{cases} 4, \text{if} & 80\% \leq performance \leq 100\% \\ 3, \text{if} & 70\% \leq performance < 80\% \\ 2, \text{if} & 60\% \leq performance < 70\% \\ 0, \text{if} & 0\% \leq performance < 60\% \end{cases} \tag{19.2}$$

The number of GP per subjects ranges from 4 (>= 80%) down to 0 (< 60%). The intention of this measure is that the maximum number n_{max} is instantiated according to (19.3):

$$n_{\max} = \begin{cases} 25 & ,if \quad GPA \geq 3.0 \\ 21 & ,if \quad 1.0 < GPA < 3.0 \\ 12 & ,if \quad GPA \leq 1.0 \end{cases} \tag{19.3}$$

It reveals that: the units a student can take are controlled by the GPA of the previous semester. The latter regulation is a consequence from the experience with students, who are obviously not able to self-estimate their capacity.

In the trade-off between a high learning quality, which is indicated by a high GPA, and a high learning quantity, which is indicated by a high number of units, some students tend to pursue the latter course at the cost of the first.

Judging from the experience of the authors, some students try to just pass examinations but don't care about the learning quality. Such an extrinsic motivation for examination results without a serious topical interest in the subject usually is revealed, when students feel that they will never need the subject's topical contents in their future carrier. Therefore, the DLNR is also a contribution to avoiding this phenomenon by enhancing the insight into the future needs of the subjects' contents, or refraining from choosing these subjects.

The introduction of the DLNRS at the school of Information Environment is supported by two subjects:

- Curriculum Planning Class: It aims at developing an individual curriculum for each student by himself that meets his needs and desires,
- Workshop: It aims at developing an ambience of mutual trust between a professor and his/her students.

The relationship among the prerequisite conditions, the GPA, the quantitative unit composition regulation for graduation, and other aspects are difficult to overview. Therefore, the development of class schedules and long-term graduation timetables is quite a challenging task.

The GPA is a mean to control the quantity (i.e., the total number of units) of subjects, but does not really help the students to choose appropriate subjects to achieve their individual goals under the terms of the complicated system as sketched out in the introduction section.

On the one hand, there is a "combinatorial explosion" of opportunities for selecting subjects (e.g., 10 out of 100 possible ones for each class period or each semester). On the other hand, there is a system of constraints, which is driven by both topical and quantitative reasons.

According to his/her individual academic history (the subjects taken in former semesters), the opportunities for a student's academic future (the subjects of the upcoming semester) are narrowed. Here, the storyboard helps to keep an overview by masking subjects, for which the prerequisites are not met.

However, practice shows that even after masking the non-options there is still a huge "search space" of possible combinations. Fortunately, some combinations require a certain number of units that might be beyond the number that is allowed by the GPA restriction rule. Here, again, the storyboard also helps by masking the non-options due to the GPA regulations.

It is really difficult to overview the "remaining choices" in a huge catalog of subjects after considering the individual combinations of topical and quantitative restrictions. Therefore, a Dynamic Syllabus system has been built, which supports the students in this complex task by way of the following four-step process:

1. Acquisition of students' data: model courses, fields after graduation, career goals, individual preferences,
2. Selection of subjects,
3. Simulation of schedules,
4. Registration of the subjects.

Where, steps 2 and 3 form a repeated process until a satisfactory solution is found, which meets all requirements and regulations.

For each subject, the Dynamic Syllabus system provides: the number of units (the number in brackets, see Fig. 19.1), the particular syllabus of this subject (in an extra window if the book symbol is clicked), and information about the prerequisites (in an extra window if the star symbol is clicked of Fig. 19.1).

Here, the creation or modification of a class schedule is a complex process of repeated prototyping and simulation with the Dynamic Syllabus system. The system ensures: prerequisites are met and there is neither a time or location conflict nor an undesired time gap in the class schedule for a semester. In Fig. 19.2 is illustrated a screen shot of the Dynamic Syllabus system in use.

Fig. 19.1 Information on a subject as displayed in the Dynamic Syllabus system

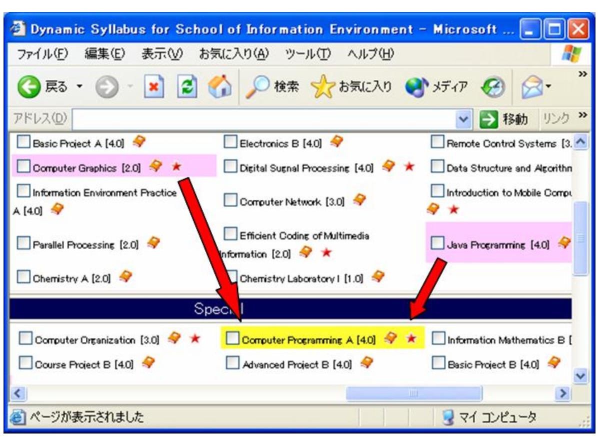


Fig. 19.2 Prerequisite conditions as displayed in the Dynamic Syllabus system

But, it does not provide an overview of the complete interdependencies and long-term schedule that meets the individual students' needs and de-sires.

In fact, the storyboard concept is a way to add this feature to the DLNRS. Our idea to improve this situation is to illustrate the interdependencies among subjects in a storyboard and thus, help the students to overview the structure of the whole subjects that they can select for their study. In this way, students can obtain an overview of the prerequisite conditions of each subject.

With a view to future trends, this approach might become consistent with the representation of the subjects' particular curriculum including the didactics of its creation. Thus, a story-board presentation here forms a top-level layer of nested storyboards for a complete academic graduation process.

Thus, we supplemented the DLNRS concept with the concept of storyboarding. For better understanding, we describe the storyboarding in the following Sect.

19.3 Storyboarding

Our storyboard is defined as follows: "A storyboard is a nested hierarchy of directed graphs with annotated nodes and annotated edges". Nodes are scenes or episodes. Scenes denote leaves of the nesting hierarchy. Episodes denote a subgraph. There is exactly one *start* and *end* node to each (sub) graph. Edges specify transitions between nodes. They may be single-color or bi-color. Nodes and edges have (pre-defined) key attributes and may have free attributes.

A storyboard may be seen as a model of an anticipated reception process that is interpreted as follows:

- Scenes denote a non-decomposable learning activity that can be implemented in any way. It can be the presentation of a (media) document, opening a tool that supports learning (URL or e-learning system) or an informal activity description,
- Episodes are defined by their sub-graph,
- Graphs are interpreted by the paths, on which they can be traversed along the directed edges,
- A start node of a (sub-) graph defines the starting point) of a legal graph traversing,
- An end node of a (sub-) graph defines the final target point of a legal graph traversing,
- Edges denote transitions between nodes. There are rules to leave a node by an outgoing edge:
  - The outgoing edge must have the same color as the in-coming edge by which the node was reached. Thus, the colors express the interdependence between incoming and outgoing edges of a node,
  - If there is a condition specified as the edge's key attribute, this condition has to be met for leaving the node by this edge.

- Key attributes of nodes specify application driven information, which is necessary for all nodes of the same type (e.g., actors and locations),
- Key attributes of edges specify conditions, which have to be true for traversing on this edge,
- Free attributes specify whatever the storyboard author wants the user to know: didactic intentions, useful methods, necessary equipment,
- The types of nodes and edges in our storyboard implementations are shown in Table 19.1 and Table 19.2.

Table 19.1 Node types of a storyboard graph

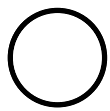
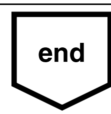
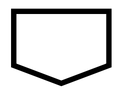
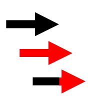
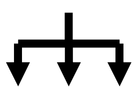
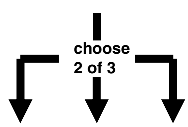
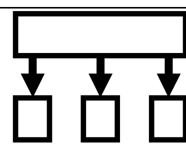
	Scene	Episode	Start Node	End Node	Reference Node
Symbol					
Interpretation	Non-composed learning activity	Composed learning activity, which is implemented by a sub-graph	Start of a graph path	End of a graph path	Re-entry point after return from a sub-graph

Table 19.2 Edge types of a storyboard graph

	Simple Edge	Fork	Fork with conditions	Alternatives
Symbol				
Interpretation	Defines a unique successor node	Defines several successor nodes, which are traversed independently in any sequence	Defines several successor nodes, which are traversed independently in any sequence, but according to the specified condition	Defines several successor nodes, out of which exactly one has to be traversed

Our storyboarding approach makes didactic design explicit, manageable, and subject to evaluation and refinement. Since the scenes are not limited to the presentation of electronic material and represent any learning activity, this concept goes beyond the IT approaches to support educational processes so far. It can be used by topical experts without an IT- or software technology background.

An essential property of the storyboarding approach is simplicity in terms of both the concept itself and the tool to implement it. Everybody, including university instructors of subjects that are far removed from information technology, is able to develop storyboards. For detailed information on the storyboarding concept, see Knauf et al. 2010.

19.4 Dynamic Storyboarding for Modeling Academic Processes

In this storyboarding application, storyboards are still of a higher level with a coarse grained granularity. Here, learning plans such as students' curriculum plans for getting their necessary academic carrier have the opportunity to be formally generated by machine support, visually checked, or dynamically verified and modified after each period of use.

More concretely, as a contribution towards utilizing the storyboard concept for an academic graduation career, we propose a storyboard, which contains courses to teach as scenes. In a later stage of this project, these scenes may turn to episodes, which need to be defined as sub-graphs by the subject matter professors for the particular courses. The final objective of this project is a complex storyboard for a university study.

Even at the top level, an easy overview can be provided by an appropriate nesting of the storyboard graphs. This overview cannot be provided by any document as commonly used in today's university practice such as lists and tables in a printed document or on a web page.

At the TDU, for example, a study of information environment is very adaptable to the students needs, but the price for this enjoyable feature is a complex system of conditions to meet, which is difficult to overview as follows:

- The study can be performed in one of three fields for a specialized subject, namely: network computing, advanced system design, media human environment design,
- Each field consists of around 60 units of major courses recommended to learn and is composed of several cores,
- The advanced system design field consists of the cores web systems, multimedia systems, and robotics,
- Two general cores, namely 2 units of orientation and 40 units of general cultural subjects, (e.g., liberal arts) are a prerequisite to graduate,
- The nesting goes even beyond that. In each core are a large number of 30-35 courses. Some of the courses belong to more than one core, others do not,
- Moreover, another 22 units of any other courses outside the selected field are required to graduate.

Does this situation call for a representation in nested structures? Yes, it does and this representation with a limited number of nodes at each level of the nesting did definitely help the students to keep an overview.

Whenever details tended to prevent students from having a general overview, we introduced an episode and shifted the details to the next lower level of the graph hierarchy.

For illustration, Fig. 19.3 shows a sub-graph of an episode called "general cultural subjects" for the undergraduate study of information environment at the TDU (see c in Fig. 19.3) and it related super-graphs (see b and a in Fig. 19.3).

However, storyboards for academic education are very individual and dynamic. The composition of a plan not only depends on general regulations, but also on individual facts like the followings:

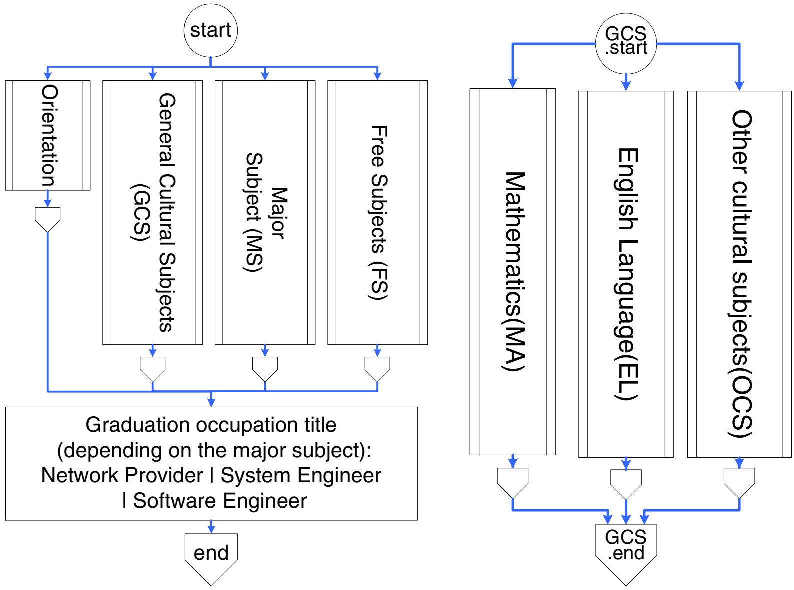
- Goals to meet: A position to reach, a company to work for, an individual talent to support, an amount of money to make,
- Preconditions to fulfill, for example:
  - A necessary pre-education,
  - A requested GPA in preceding semesters or of the school-leaving examinations,
  - Necessary equipment like a notebook for programming seminars or a car to reach the university campus,
  - A certain language skill to understand topical contents,
  - A necessary amount of resources such as money or time,
  - Talents to address such as being creative, analytic, sportive, or having certain social skills.

In the undergraduate study of the TDU students, for example, there is the Grade Point System that limits the number of units to take for an upcoming semester as detailed in the previous section. This definitely narrows the storyboard for an upcoming semester, depending on the results of the previous one, namely, individual higher level storyboards of a particular student is very dynamic.

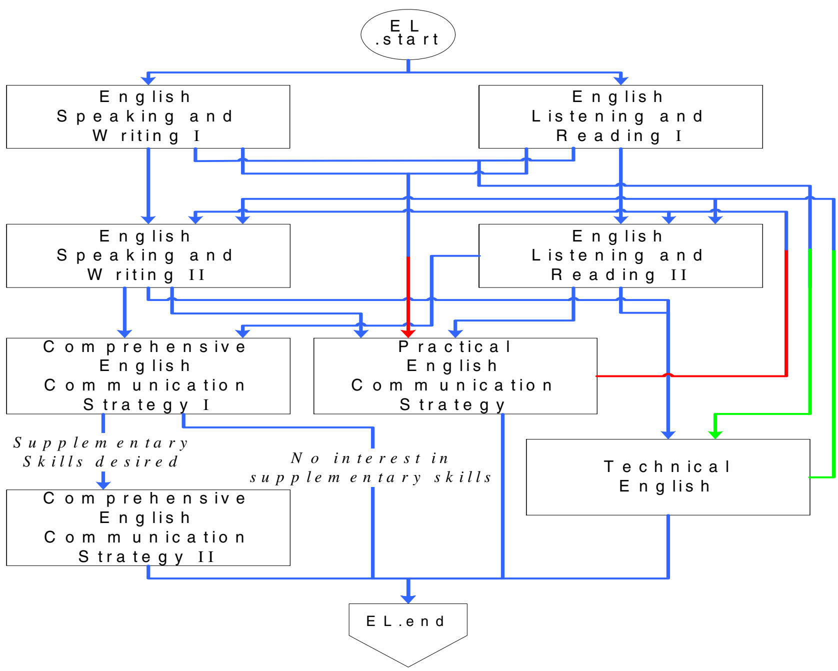
Since the rules for this dynamics are known and fixed, we were able to implement them into the system and thus, to ensure that these rules are met by each student. The curriculum plan is subject to changes after each semester not only according to their results of the previous semesters, but also according to changing objectives, newly revealed talents or interests or new course offers. Therefore, the approach is called dynamic storyboarding.

Also, individual learning plans should not only be based on individual quantitative capability like a GPA of former students who went through similar ways. Additionally, individual properties, talents, and preferences should be considered.

For example, some students are more talented for analytical challenges, some are more successful in creative or composing tasks, and others may have an extraordinary talent to memorize a lot of factual knowledge.



(a) Super-graphs of (b) (b) Super-graphs of (c)



(c) Graphs of English subjects

Fig. 19.3 English subjects of Tokyo Denki University

Consequently, we had to include a student profile to avoid lavishing suggestions upon the students, which do not match their individual preferences and talents in this project. Before we present our profiling concept, we introduce the EDM technology to dynamically validate and refine students' university curricula.

19.5 Intelligent and Adaptive Curriculum Mining

A basic objective of storyboarding is to use knowledge engineering technologies on the (semi-) formal process models. In particular, we aim at inductively "learning" successful storyboard patterns and recommendable paths. It is performed by an analysis of the paths where former students went through the storyboard. To show the feasibility and benefit of high level storyboarding for its qualified assistance of students, we developed a prototype storyboard for curricula of a university study. This prototype is used to validate curricula created or modified by the students in advance of their study (Sakurai et al. 2009).

For such a purpose, we introduced a concept to estimate achievable GPA of curricula, which are composed by university students in their curriculum planning class in the first semester. For such curricula we developed an EDM technique, which is applied to storyboard paths that (former) students went. Based on these examples, the success in terms of GPA of intended paths can be estimated. This technique is applied to the paths of students through a storyboard, which shows possible ways through a complete study.

We looked for model that only contains the available data and no derived assumption or interpretation of it. Course sequences are, formally spoken, "words" (paths of educational activities) of a formal language, which contains the courses as its alphabet.

Deriving nodes of a Bayesian Belief Network or an inference diagrams with Markov Decision Processes (MDP) (Lacave et al. 2007, Puterman 1995) for example, along with edges that express interdependencies between the nodes, would be a interpretation of the available data by deriving attributes and finding statistical interdependencies in-between them. We avoided that on purpose and composed decision trees out of the paths to apply EDM on the original data only without making an interpretation of it.

To construct a DT other methods are also proposed. However, the decision required here is neither fuzzy, nor binary, or multivariate decision (Wei-hong et al. 2010, Cha and Tappert 2009, Amasyali and Ersoy 2005. Further, construction and search efficiency is not so important since the construction is done incrementally every year for only the year's graduated students and the query is done by crisp matching between groups of subjects studied each semester in series.

Thus, such a simple and crisp matching type DT construction heuristics required here is newly proposed, which is fit to curriculum suggestions for obtaining an estimation of success in terms of GPA along with a recommended refinement of the curriculum with respect to GPA optimization.

19.5.1 Data Preprocessing

In a pre-processing step to determine the paths, the individually visited items (episodes and scenes) in the storyboard are "flattened down" to a big graph that contains scenes only. This is performed by systematically replacing episodes by the individually visited items of the episode's related sub-graph. In the granularity of this storyboard application, a scene is a course that holds over one semester.

To form the decision tree, the path needs to be flattened to a sequence that contains scenes only. This is performed by a function like this:

```
input: pointer to the first node of a storyboard path
  that contains all types of nodes
function value: pointer to the first node of a storyboard
   path that contains scenes only
function flatten(input)
if input = start_node then flatten := start_node
if input = end_node then flatten := end_node
if input = scene then begin
   output := scene;
   %recursive call with next node
   output.next_node:= flatten(input.next_node)
End {if};
if input = episode then
     output.next_node:=
    flatten(first_after_start_node(episode_sub-graph))
     % hook in the result of the recursive call with the
     episode's sub-graph
End {flatten};
```

Fig. 19.4 shows a small example storyboard path, which a particular student went through, along with the result of the flattening procedure. This student finalized his study with a success level (more concretely the GPA) of 3.0. It means *A* in the letter-based evaluation system. On the left-hand side of Fig. 19.4, the path is in the nested form as derived from the storyboard, which is a nested graph, too.

19.5.2 Construction of the Decision Tree

The construction of the decision tree is based on the paths of former students through the storyboards which model the "space of opportunities". In the space, each of the students took a particular one, namely, a path through the storyboard. After the flattening procedure mentioned above, the paths are available as sequences of atomic nodes or sets of atomic nodes and end with a GPA label node.

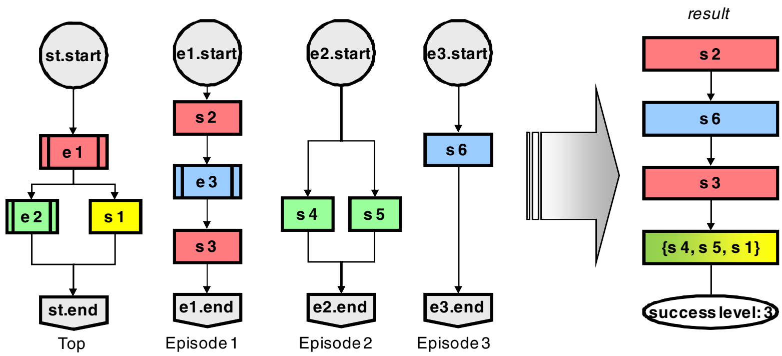


Fig. 19.4 A student's path through a nested storyboard is "flatten down

Since it may happen that nodes of different sub-storyboards carry identical names (like *Exercise* # 1, *Example*), the nodes in the storyboard hierarchy must receive unique names. This has been performed by adding the upper episode names as a prefix and separating the prefixes by a dot. For instance: *e2.e5.s1* is the new name of scene *s1* in episode *e5*, if *e5* is in the sub-storyboard of episode *e2*.

The DT is based on the concept of bundling common starting sequences of the various paths to a node of the tree. In Boeck 2007, these starting sequences are called "least common denominator". Of course, all paths traversed by the students begin with the start node that forms the root of the DT.

Different subsequent following (*next*) nodes of the paths will result in different sub-trees right below the actual root on the last node of the common starting sequence. This continues for each lower level sub-tree accordingly. If there are several paths with a common starting sequence from the root to the actual root, which is different in the next (subsequent) nodes, related sub-trees will be established.

The DT construction is performed as follows:

- The start node of the top level storyboard forms the root,
- From there, all paths are partitioned according to their next element in the scene (or scene set) sequence,
- For each "next element" a related sub-tree is added to the considered node,
- This is done recursively for the sub-trees until all paths are included.

The final nodes of the paths are followed by a label-node. Label-nodes contain a list of marks that students received after going through this path. Each mark is along with the number of occurrences (the number of students getting the mark). Additionally, this label is attached to the weighted arithmetic average (WAA) value of these marks, too. The value of WAA serves as an estimate of GPA for future students who plan to go through the same path. For efficient computation, the sub-trees of each non–leaf node are sorted from left to right by ascending root nodes (that contain a code of the related course). The DT construction is performed by a function like this:

```
input: • path: a pointer to the 1st node of a storyboard
         path went by a student
       • tree: pointer to root of the decision tree so far
        (initially empty)
function value: • pointer to the decision tree that
         incorporates the new path
function create_tree(path,tree)
 if tree = empty then create_tree := path
     % initial tree := 1st submitted path
 else begin
  if tree.content = path.content then begin
   i := 1;
   while tree.sub_tree(i).content < path.content
    i:=i+1 % ignore sub-trees with roots < 1st path node
    if tree.sub_tree(i).content = path.next_node.content
     create_tree(path.next_node,tree.sub_tree(i))
     % recursive call with next node and related sub-tree
    else begin
     n := number_of_subtrees; k := i;
     for j:= i+1 to n+1 do begin
      tree.sub_tree(j) := tree.sub_tree(k);  k := k+1
      % shift remaining sub-trees one position place to
       right
     end {for};
     subtree(i) := path
     % hook in the path as a new sub-tree at position i
    end {else};
 end{else}
end {create_tree};
```

Fig. 19.5 depicts the result of the DT construction for a setting. At left-hand of Fig. 19.5, 15 students went through the storyboard on four different paths, namely:

- [s4, {s6, s7}, s1, s9], depicted by red background, 4 students passed along,
- [s4, {s6, s7}, s5, s8], shown by yellow background, 5 students passed along,
- [s4, s2, {s3, s1, s5}, s9], stated by green background, 5 students passed along,
- [s4, s2, {s3, s1, s5}, s6], pointed by blue background, 2 students passed along.

In the derived decision tree as illustrated on the right-hand side of Fig. 19.5, each of these four paths form a path in the tree from the root towards a leaf. Attached to each leaf, there is a label node, which holds the GPA information of the students, who passed along this path, namely:

- The average (the GP achieved by the students, weighted by the number of students with this result)
- The particular distribution of the different GP.

[s4, {s6, s7}, s1, s9]		[s4, {s6, s7}, s5, s8]	
4	*1 student*	4	*3 students*
3	*2 students*	3	*0 student*
2	*1 student*	2	*2 students*
0	*0 students*	0	*0 students*
[s4, s2, {s3, s1, s5}, s9]		**[s4, s2, {s3, s1, s5}, s6]**	
4	*3 students*	4	*0 students*
3	*2 students*	3	*0 students*
2	*0 students*	2	*1 student*
0	*0 students*	0	*1 student*

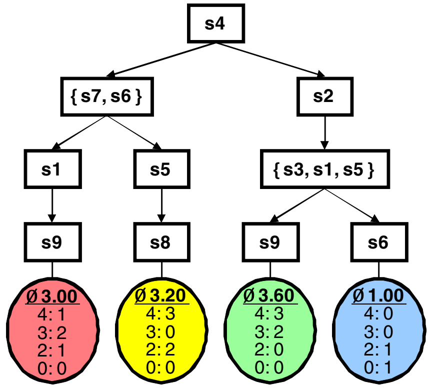


Fig. 19.5 Storyboard paths and a derived decision tree

Since the courses of a semester are usually visited concurrently (represented by the fork edges of Table 19.2), they are considered as a single node in the DT containing a set of courses.

A new path is added to the tree by simultaneously traversing the path's courses sequence and the DT down from the root until the path is finished or there is a "next node" in the path that is different from all "next sub-tree roots".

In the first case, the related new label node will be updated accordingly or added to the knowledge base, if there is no one so far. In the latter case, a new subtree is made out of the remaining path and hooked into the tree.

Based on this information, the student makes a decision on whether or not holding on to the submitted curriculum or modifying it in accordance with the optimal supplemental path.

19.5.3 Utilizing the DT for Curriculum Assessment

If a submitted path is completely represented in the DT as a path from its root to a node that is succeeded by a label node (i.e., with an assess - fact), the GPA estimation is very easily done through presenting the content of this label.

In the other case, namely, if a student submits a curriculum plan that is not represented in the DT, the most similar sub-path in the DT is identified.

In our initial approach, similarity refers to the number of same course sets in sequence, which the path has in common with a path represented in the tree. This similarity measure s is in the range $0 \leq s \leq 1$. In the worst case, there is no node in common with any path in the tree (i.e., similarity $s = 0$) and in the best case, the submitted path is completely represented in the DT (i.e., similarity $s = 1$).

Like in the DT build procedure, this is made by simultaneously traversing the path's course sequence and the DT down from the root until the path is finished or there is a *next* node in the path that is different from all *next* sub-tree roots.

In the first case, the related label node at its leaf position provides the desired GPA estimation. In the latter case, the label node of the current tree position provides the desired information. Of course, in such a case, one may be interested in suggestions to modify the submitted path in a way that the chances for a higher GPA reach an optimum or become the highest in their value.

Thus, for example, it is suggested to exchange the submitted remaining path for the most successful alternative remaining path with the best WAA value among the remaining paths in the DT whose common leading part is the longest.

For this purpose, we supplement the estimated GPA with the most successful rest - path starting at the last node of the tree traversing. We provide this optimal supplement along with its achievable GPA, if this optimum is better than the GPA estimation of the submitted path.

Also, the user is informed of the degree of similarity of his submitted path and the one found in the DT. We call this similarity significance and compute it as the number of nodes in sequence that are common in the submitted path and the DT, related to the entire length of the path.

Based on this information, the users (students) can make a decision on whether or not they should hold on to the submitted curriculum or modify it in accordance with the optimal supplemental path. More concretely, if a student submits a path, which he/she planned in the curriculum planning class, the path's GPA estimation by using the DT is performed as follows:

- If the submitted path is completely represented in the tree,
  - The GPA estimation will be given by providing a GPA label (weighted GPA and their distribution),
  - The significance of this estimation is 1, because it is based on information of the complete submitted path,
  - The recommended rest path is empty, because the DT does not contain a supplemental path, which leads to the improvement in success in terms of an optimal GPA.
- If the submitted path is partly represented in the DT,
  - The GPA is computed by merging the GPA labels of all sub-trees starting from the last node that have both the submitted path and a related path in the tree in common,
  - The significance is computed as the number of common nodes divided by the total number of nodes in the path,
  - The recommended rest path is the best rated path in the tree after the end node of submitted path.

A procedure to compute the GPA estimation, its significance and a rest path recommended from the point, from which the submitted path differs from and path in the DT is like follows. Once more, for efficient computation, the sub-trees of each non–leaf node are sorted from left to right by ascending root nodes that contain a code of the related course.

```
input:  • path: a pointer to the first node of a storyboard
          path went by a student
        • tree: pointer to root of the decision tree
output: • estimated GPA and their distribution with former
          students its significance
         • pointer to the recommend rest path from the
          point where the submitted paths differs from the
          paths represented in the tree

procedure use_tree(path,tree,GPA,significance,restpath)
begin
    use(path,tree,GPA,restpath,1,depth);
    % depth is the length of the subpath that is depícted
      in the tree, its initial value is 1 significance
      := depth / length(path)
end {use_tree};

procedure use(path,tree,GPA,restpath,depth,total_depth)
begin
  if tree.content = label_node then begin
     % path is traversed and was completely in the tree
     GPA := tree.content;
     rest_path := empty;
     total_depth:= depth
  end {if};
  if tree.content = path.content then begin
     % the actual path node it in the tree
     i := 1;
     while tree.sub_tree(i).content < path.content
      i := i+1; % ignore sub-trees with roots<1st path node
      if tree.sub_tree(i) = path.next_node.content then
         % next node is in the tree, too
         Use(path.next_node,tree.sub_tree(i), GPA,
             restpath, depth+1,total_depth)
      else begin
       % from here, the path is not represented in the tree
       GPA := merge(GPAs_of_all_subtrees_from_here);
       restpath := path_with_best_GPA_downward_from_here;
       total_depth := depth
      end {else}
  end{if}
end {use};
```

Fig. 19.6 shows the usage of the DT for three submitted paths for a simplified example setting. There is one path, which is represented completely in the DT (indicated by green color) and two paths, which are not represented completely in the DT (indicated by blue and red colors).

The GPA estimation of the green path is simply performed by providing the related GPA label of the related path in the DT. For the blue path, there is no identical path in the tree. Here, the estimation procedure looks for a path within the tree, which has the longest starting sequence in common with the submitted path. This is [s4, {s7, s6}], in our example. Since this path has only two nodes in common with the submitted one (having four nodes), the significance of the GPA estimation is calculated by 2/4, i.e. 0.5.

For the blue path, there is no identical path in the tree. Here, the estimation procedure looks for a path within the DT, which has the longest starting sequence in common with the submitted path. This is [s4, {s7, s6}], in our example. Since this path has only two nodes in common with the submitted one (having four nodes), the significance of the GPA estimation is calculated by 2/4, i.e. 0.5.

Behind the node {s7, s6}, there are two different sub-trees, which led to different success degrees in terms of GPA that have been achieved by former students, [s1, s9] and [s5, s8]. Since the latter is the better one, it is recommended as a rest path to optimize the achievable GPA.

For the red path, the usage of the decision tree is performed accordingly.

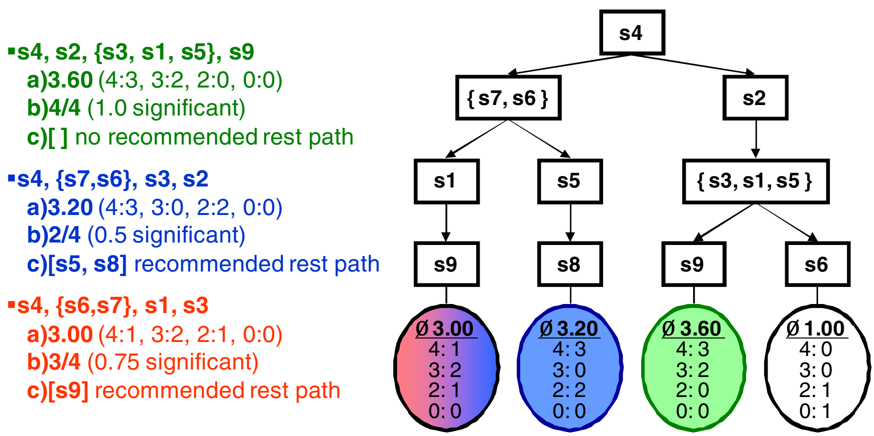


Fig. 19.6 Examples for (a) success estimation, (b) its significance, (c) recommended rest paths

19.5.4 Personalization by Student Modeling

Our mining results would be more significant, if individual properties, talents, and preferences are considered as well. For example, some students are more talented for analytical challenges, others are more successful in creative or composing tasks, and some may have an extraordinary talent to memorize a lot of factual knowledge. Consequently, we need to include individual student profiles to avoid lavishing upon the students, suggestions that do not match their individual preferences and talents, which are derived from their educational history and additional data that describes individual characteristics.

Unfortunately, we could not obtain sufficient data to outcome this kind of learner profiles. Therefore, we refrained from deriving an explicit model as we did in the past by deriving estimating intellectual traits from Gardener 1993 and learning styles from Felder and Silverman 1988.

In Knauf et al. 2009, we presented an approach of personalized EDM by deriving estimating intellectual traits from Gardener 1993 and learning styles from Felder and Silverman 1988. Once again, we could not obtain sufficient data to set this kind of learner profiles. Thus, we gave up from deriving an explicit model.

Instead, we shifted strategy from an "eager" one of holding an explicit model towards a "lazy" strategy of mining with data, which is really available, holds empirically, and is not a result of "guesses" about the students' general characteristics. In particular, we utilize the educational history of the students and vocational ambitions for student modeling. This new issue is the focus of this Sect.

19.5.4.1 Modeling Educational History

As educational data of a student, we have for each completed semester the next information:

- The subjects taken in it,
- Its number of units,
- The rating in it reached by the student, such as: *S* for > 90%, *A* for 80-89 %, *B* for 70-79%, *C* for 60-69%, *D* for 40-59%, *E* for < 40%,
- The related GP as follows: 4 for *S* and *A*, 3 for *B*, 2 for *C*, and 0 for *D* and *E*.

In order to model the educational history of a student, we form a set of pairs by means of (19.4), where s = $\{[s_1, l_1], [s_2, l_2]\ldots [s_n, l_n]\}$ being the set of codes for subjects taken by the student, l_i being the success level in terms of GP of the student in subject with the code s_i defined as the number of GP received for the related rating in the Japanese system[1] (e.g., the rating in the Japanese rating system is: *S*, *A*, ..., *E*), as it is explained in (19.5)

$$p = \{[s_1, l_1], [s_2, l_2], \cdots, [s_n, l_n]\} \tag{19.4}$$

$$l_i = \begin{cases} 4, \text{if the student reached rank } S \text{ in subject } i \\ 4, \text{if the student reached rank } A \text{ in subject } i \\ 3, \text{if the student reached rank } B \text{ in subject } i \\ 2, \text{if the student reached rank } C \text{ in subject } i \\ 0, \text{if the student reached rank } D \text{ in subject } i \\ 0, \text{if the student reached rank } E \text{ in subject } i \end{cases} \tag{19.5}$$

[1] *S* and *A* as well as *D* and *E* are not distinguished in terms of Grade Points. *D* and *E* make a difference only in terms of the acceptance of a subject as a prerequisite of another subject.

To include pre-university education as well as diverse other available data, which may characterize a student, we include four subjects that are part of pre-university examinations, namely:

- Mathematics, Science (mostly Physics),
- English and Language Arts (Japanese),
- Three subjects that might characterize a student, too: High School Recommendation, Self Recommendation, and some special skill, which is proven and checked by the university administration.

19.5.4.2 Modeling Vocational Ambitions

For the self-estimation of vocational ambitions, the student is asked about his/her preferred field in his/her future job. The student is asked: What kind of job do you prefer? Namely: 1) topical work; 2) administrative work; 3) research work; 4) teaching work. The student can select one (more than one, if appropriate) of these categories. Then, the student selects topical fields, in which he/she looks for such a position as characterized above. This is performed in 2 two level manner. Student is requested to rate on a scale ranging from 0 to 4.

According to the faculty structure at the School of Information Environment of the TDU, they are firstly asked to rate three major fields. For those of these major fields, which received a rate different from zero, student is asked to rate more precisely among three subfields. As a result, we have up to 13 pairs [*ambition*, *level*], namely up to 4 characteristics and up to 9 topical fields.

19.5.4.3 The Derived Student Model

The student profile is a vector of pairs [*profile item*, *level*], namely up to 5 pairs [*subject*, *success level*] that depict the educational history, and up to 13 pairs [*vocational ambition*, *level*] to reveal vocational ambitions. All these pairs have the same range of levels, where the levels range between 0 and 4 for each profile item.

After some learning success results (GP for subjects) become available, namely after each semester, the profile is extended by the results (ratings) in the courses the student has taken so far, and gains more and more items after each semester according to the number of courses the student has taken.

For a personalized GPA prediction of curricula, we have to determine a degree of similarity between the educational history of a student under estimation and that of the students in our data base. Similarity of profiles means similar relations in-between its components, which is expressed by the Cosine Coefficient.

Here, the profiles are vectors of success levels and the cosine coefficient between two vectors $X = [x_1, x_2 \ldots x_n]$ and $Y = [y_1, y_2 \ldots y_n]$ is estimated by (19.6)

$$\cos(X,Y) = \frac{\sum_{k=1}^{n} x_k y_k}{\sqrt{\sum_{k=1}^{n} x_k^2} * \sqrt{\sum_{k=1}^{n} y_k^2}} \tag{19.6}$$

Considering each profile a 178 dimension vector (all existing courses, the pre-university education items, the intellectual trait items, and the vocational ambition items) is not appropriate for space and time complexity reasons.

The aim to build a vector with a dimension of the cardinality of the set of pairs $p| = |\{[s_1, l_1], [s_2, l_2], ..., [s_n, l_n]\}| = n$ [2] for each student profile is not appropriate. One the reason is: two such vectors, for which the similarity has to be determined, are likely to have a different number of dimensions.

Thus in order to determine an appropriate similarity measure *sim* (p_1, p_2) for two profiles $p^1 = \{[s_1^1, l_1^1], [s_2^1, l_2^1], \cdots, [s_n^1, l_n^1]\}$ and $p^2 = \{[s_1^2, l_1^2], [s_2^2, l_2^2], \cdots, [s_m^2, l_m^2]\}$ form two k dimensional ($k \le m + n$) vectors $\vec{p}^1 = [l_1^1, l_2^1, \ldots, l_k^1]$, $\vec{p}^2 = [l_1^2, l_2^2, \ldots, l_k^2]$ by means of:

- Merging the subject sets $s^1 = \{s_1^1, s_2^1, \cdots, s_n^1\}$ and $s^2 = \{s_1^2, s_2^2, \cdots, s_m^2\}$ towards $s = s^1 \cup s^2 = \{s_1, s_2, \cdots, s_k\}$ with $k \le m + n$,
- Adopting the l_i^j from the original profiles, if subject s_i^j is an element of p^j, i.e. $s_i^j \in \Pi_1(p^j)$), [3]
- Setting $l_i^j := 0$, if subject s_i^j is not element of p^j, i.e. $s_i^j \notin \Pi_1(p^j)$,
- Using the Cosine coefficient as similarity measure $sim(p^1, p^2) = \cos(\vec{p}^1, \vec{p}^2)$.

In our new approach that includes the student profiles, we construct the DT exclusively from students with profiles that are most similar to the one under evaluation.

To compose the subset $s^{sim} \subseteq s$ of most similar students to a student under evaluation with a student profile p^{eval}, we state a portion (a percentage *prc*) of students, whose paths are most similar to the submitted ones.

In this way, the estimation of success chances in terms of a likely to achieve GPA is based on individual preferences, talents, and weaknesses. In addition, the suggestion of a remaining curriculum path (subjects recommended to optimize the success of study) is adapted to individual properties, because it is calculated on the base of examples with a similar profile. Both processes are made by applying the technology to groups of students with similar attributes, behaviours, outcomes only.

In contrast to our former profiling approach (Knauf et al. 2009) of detecting intellectual traits (Gardner 1993) and learning styles (Felder and Silverman 1988), this profiling approach is quite dynamic and improving over time. Subjects currently taken will be history in the next semester and their results are useful information for a more precise profiling. Therefore, it is essential to repeat the GPA estimation after each semester and include the newest available data.

---

[2] $|M|$ is the cardinality of a set M, i.e. its number of elements.

[3] $\Pi_1(M)$ with $M = \{[e_1, \cdots], [e_2, \cdots], \cdots, [e_n, \cdots]\}$ being a set of vectors denotes the projection of each vector's 1st element into a set: $\Pi_1(M) = \{e_1, e_2, \cdots, e_n\}$.

In conclusion, this technology is adaptive in terms of computing the mining results due to both the educational history of the considered student and the data base whose information is dynamically updated by the students' study results for the DM technology.

19.5.5 Evaluation of the Approach

The evaluation is a quite complex procedure. To have a sufficient amount of data of former students, for which we could not collect all profile data, we used only data that is available for all students (i.e., the learning results at the university). The more profile items we include, the more the students are diversified (i.e., the derived model tends under fit).

We collected 188 individual storyboard paths of students, who studied Information Environment at the School of Information Environment of the TDU from 2005 till 2009. Unfortunately, these students left the university before we developed the current profiling concept. So, as a profile, we could not include the vocational ambitions and the educational history is limited to the university history and does not include pre-university education.

With the aim to include the complete profile data of the current approach, we needed to collect a sufficient amount, which takes several years. Of course, we started collecting it from the moment we came up with the current approach.

From these samples, we removed two samples of students, who entered the university after spending several semesters elsewhere, because their marks were derived by recognition of marks received in similar subjects at another university. This led to 186 samples.

After collecting and studying all the samples and organizational material rules to compose a curriculum, we chose a compact data representation by coding the particular subjects and the particular students. By using subject codes 1-155 and student IDs 1-186, we composed a complete DT from the 186 samples.

To make sure that identical starting sequences of semester curricula really end up in the same path, the DT is well sorted as follows:

1. The subject sequence within a semester is sorted by ascending subject codes,
2. The students' samples are sorted by the code lists, which are compared element by element, ascending, too.

We adopted this technology from a similar technology, which is usually performed in DM for item lists to efficiently generate association rules. Fig. 19.7 shows an extract of the DT composed by all the samples. For each student (coded by his/her ID) according to:

- Each semester (columns *s*, with yellow-brown background),
- The subjects (courses, columns *c* with light green background),
- Their number of units (columns *u* with light yellow background),
- The achieved results (with light blue background), the mark (columns *m*: *S*, *A*, *B*, *C*, *D*, or *E*) and the number of grade points (columns GP: 4, 3, 2, or 0) are listed up.

ID	s	c	u	m	GP	s	c	u	m	GP	s	c	u	m	GP	s	c	u	m	GP	s	c	u	m	GP	s	c	u	m	GP	s	c	u	m	GP	s	c	u	m	GP	GPA
5	1	11	3	A	4	2	29	4	A	4	3	21	2	B	3	4	9	2	C	2	5	10	4	A	4	6	13	4	A	4	7	1	4	A	4	8	2	4	A	4	**3,48**
		17	4	B	3		49	4	S	4		30	4	A	4		14	2	A	4		12	4	A	4		20	2	A	4		84	2	S	4						
		26	2	B	3		92	4	C	2		32	3	C	2		35	3	B	3		14	2	A	4		70	2	S	4											
		36	1	A	4		96	3	C	2		50	4	A	4		41	3	S	4		19	2	A	4		105	2	A	4											
		58	2	A	4		116	3	A	4		57	3	S	4		64	3	C	2		87	3	B	3		140	3	A	4											
		94	2	B	3		130	2	A	4		73	3	B	3		75	3	B	3		99	2	A	4		153	2	B	3											
		129	2	C	2							148	2	B	3		82	3	B	3		120	3	S	4																
		155	1	S	4												141	2	B	3		124	2	A	4																
157	1	11	3	S		2	29	4	S	4	3	21	2	A	4	4	9	2	B	3	5	10	4	A	4	6	13	4	A	4	7	1	4	A	4	8	2	4	A	4	**3,72**
		17	4	A	4		49	4	A	4		30	4	C	2		35	3	B	3		12	4	A	4		70	2	A	4											
		26	2	A	4		92	4	S	4		32	3	C	2		41	3	S	4		19	2	A	4		79	3	A	4											
		36	1	A	4		96	3	A	4		50	4	A	4		64	3	A	4		24	2	B	3		140	3	S	4											
		58	2	C	2		116	3	A	4		57	3	S	4		75	3	B	3		63	3	A	4		152	2	A	4											
		94	2	B	3		130	2	A	4		73	3	A	4		82	3	B	3		87	3	A	4		153	2	B	3											
		129	2	A	4							148	2	A	4		141	2	A	4		120	3	S	4																
		155	1	A	4												143	2	A	4																					
47	1	11	3	A	4	2	29	4	B	3	3	30	4	C	2	4	9	2	C	2	5	10	4	B	3	6	13	4	S	4	7	33	4	S	4	8	34	4	S	4	**3,31**
		17	4	A	4		49	4	S	4		32	3	B	3		35	3	C	2		12	4	A	4		70	2	B	3		84	2	S	4						
		26	2	A	4		92	4	C	2		50	4	A	4		41	3	A	4		19	2	A	4		79	3	A	4											
		36	1	A	4		96	3	S	4		57	3	S	4		64	3	C	2		63	3	B	3		140	3	A	4											
		58	2	A	4		116	3	B	3		73	3	A	4		75	3	C	2		87	3	A	4		152	2	B	3											
		94	2	C	2		130	2	A	4		111	2	B	3		82	3	C	2		120	3	A	4		153	2	B	3											
		129	2	A	4							148	2	B	3		141	2	D	0		124	2	B	3																
		155	1	A	4												143	2	B	3																					
56	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	**3,90**

Fig. 19.7 Extract from the decision tree data as an Exel-Sheet

The last row contains a weighted (by the number of units) GPA, which quantifies the degree of success in the study. Again, both the subject lists of the students within a semester and the complete students' samples (which are lists of lists) are sorted by subject code.

The bars between the paths show, up to which semester the curricula of adjacent students are identical (circles) respectively from which semester they are different from each other (bullets). Thus, the grey bars separate the sub-trees from each other.

The entire table has 42 columns and 1616 rows. Figuratively speaking, the table illustrates the DT in a horizontal direction with the root on the very left-hand side and the leaves on the very right-hand side. The grey bars separate sub-trees from each other.

To evaluate this approach, we used these 186 individual storyboard paths and validated the approach by cross-validation with a subset size of 1 (i.e. the so called leave one out approach). In a number of cycles equal to the number of samples, we removed one path from the complete DT and used this sample to check the remaining DT.

As a result, we received a list of all the samples along with their original (real) GPA, and the GPA as estimated by the EDM technology as shown in Table 19.3.

The proposed system just suggests the successful curriculum by refining a proposal submitted by a student with respect to best possible results according to data of former students, who went similar way through the storyboard and had similar grade points so far.

Namely, the curriculum suggestion by this system is not protected in that a student can neglect the suggestion by his self-realizing prophecy or that of his leading teachers. Since such prophecy is subjective and may sometimes be significantly wrong, the neglect should be done at student's risk or at his teachers' risk.

Further, the success chance is fundamentally measured as average GPA of students traversing the same path recognized through matching subject (or course) sequences. Thus, the minor changes can be reflected or mitigated every year by mining the newly graduated students' data. In case of major changes such as a change of a teacher, DM through extracting such old data as including those of classes or subjects taught by past teachers may be useful in the beginning, but stepwise outdated the more data about the new teacher is available.

Table 19.3 Validation results

Student ID	GPA	GPA estimated by DM	Difference
89	3.40	3.23	0.17
179	3.30	3.24	0.06
92	3.55	3.63	0.08
164	3.29	3.71	0.42
177	3.52	3.60	0.08
...	...	...	...

A histogram which illustrates the complete data is shown in Fig. 19.8. Through the visual observation of the graph in Fig. 19.8, the difference between real and estimated GPA is found small. Statistically the mean of the difference between both was 0.43 with a standard deviation of 0.30. These show the estimation by our DM method is mostly correct and useful for estimating the GPA and suggesting near optimal curriculum refinements.

Fig. 19.8 shows the complete validation results. Each marked point at the *x*-axis denotes a student and the *y*-axis his/her GPA. The black points represent the real GPA achieved by the student. The red points represent the GPA that would have been estimated by our EDM technique based on the data of the other students.

Of course we wanted to know which circumstances promise a good estimation and which do not. Unfortunately, due to data privacy protection, we did not have other data of the students such as: age, sex, or family status; although all the data was anonymous (each student was just a number in our study).

The only thing we could analyze is, whether or not there is a correlation between the proficiency level (GPA) and the quality of the results of our technology. We expected the technology to be worse in low level GPA and becoming better and better with increasing GPA, but got a surprising result.

As shown in Fig. 19.9, our expectation about low level GPA students was apparently not wrong and we mistook in both directions (i.e., over- and underestimated their GPA). For the majority of students, who have a GPA around three, we achieved mostly good results. But for very high proficiency students with a GPA of 3.5 or higher (i.e., these were 37 out of the 186 students) the risk that our estimation is far from the truth, was growing again.

Currently, we are discussing, what additional data we could include into our EDM technology to improve the result especially for this kind of students.

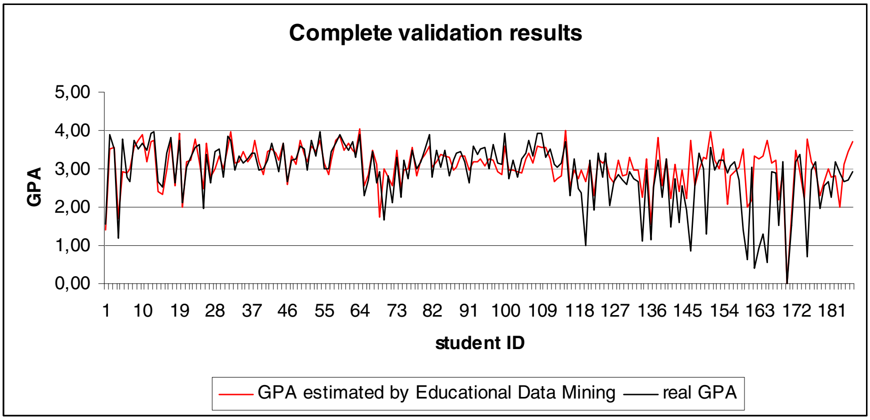


Fig. 19.8 Complete validation results

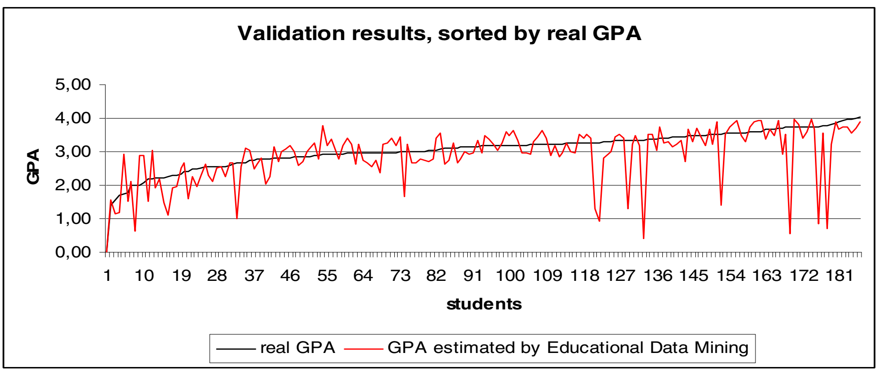


Fig. 19.9 Complete validation results, sorted by GPA

19.6 Conclusions

The chapter introduced an intelligent system for modeling and supporting academic educational processes. The intelligence of the system is performed by Knowledge Engineering methods such as EDM to a semi-formal model of the process by means of storyboarding. In this way, optimal didactic success patterns with proven excellence can be inferred, which may have never been revealed by conventional (non-intelligent) methods of analyzing students' educational data.

It consists of mining a DT and applying this DT to curricula planned by current students. Thus, students receive an estimation of the GPA along with a recommendation to supplement it to reach optimal success.

This technology is adaptive in terms of computing the mining results due to both the educational history of the considered student and the data base whose data is dynamically updated by the students' study results for the DM technology.

Also, an approach to include individual learner profiles was introduced. The profiling concept initially uses the per-university educational history and is dynamically extended by the students' university study results. In this way, the students are grouped into groups with similar attributes, behaviors, and outcomes. The original general (non-individual) method, namely, the proposed DM method, is applied to students with profiles of a high degree of similarity to the student under consideration.

A feasibility study showed its usefulness of the system. It supported students' dynamic learning activities by features that help to overview, verify, refine, and optimize their own individual curricula. The effect has been validated by cross-validation with about 200 students' records. The mean of the difference between the original GPA and the estimated one was 0.43 with a standard deviation of 0.30.

References

Amasyali, M.F., Ersoy, O.: Cline: New multivariate decision tree construction heuristics. In: Proceedings of CIMA (2005)

Boeck, R.: Ein data mining verfahren zur pfadbewertung in storyboards. (German) Diploma thesis, Ilmenau University of Technology (2007)

Cha, S., Tappert, C.: A genetic algorithm for constructing compact binary decision trees. Journal of Pattern Recognition Research 1, 1–13 (2009)

Dohi, S., Nakamura, S.: The development of the dynamic syllabus for school of information environment. In: Proceedings of ITHET, pp. 505–510 (2003)

Felder, R.M., Silverman, L.K.: Learning and teaching styles in engineering education. Engineering Education 78(7), 674–681 (1988)

Gardner, H.: Frames of Mind: The Theory of Multiple Intelligences. Basic Books (1993)

Knauf, R., Sakurai, Y., Takada, T., Dohi, S.: Personalized Curriculum Composition by Learner Profile Driven Data Mining. In: Proc. of the 2009 IEEE International Conference on Systems, Man, and Cybernetics (SMC 2009), San Antonio, TX, USA, pp. 2137–2142 (2009) ISBN 978-1-4244- 2794-9

Knauf, R., Sakurai, Y., Tsuruta, S., Jantke, K.P.: Modeling Didactic Knowledge by Storyboarding. Journal of Educational Computing Research 42(4), 355–383 (2010)

Lacave, C., Luque, M., Díez, F.J.: Explanation of Bayesian networks and influence diagrams in Elvira. IEEE Transactions on Systems, Man and Cybernetics, part B: Cybernetics 37, 952–965 (2007)

Rexer, K., Allen, H., Gearan, P.: Data miner survey. In: Proceedings of PAW (2010)

Puterman, M.L.: Markov decision processes: Discrete stochastic dynamic programming. Wiley, New York (1995)

Sakurai, Y., Dohi, S., Tsuruta, S., Knauf, R.: Modeling academic education processes by dynamic storyboarding. Journal of Educational Technology & Society 12, 307–333 (2009)

Weihong, W., Wei, R., Qu, L.: Fuzzy decision tree construction with gene expression programming. In: Proceedings of ISKE, pp. 244–248 (2010)

Abbreviations

DLNR	Dynamic Learning Need Reflection System
DM	Data Mining
DT	Decision tree
EDM	Educational Data Mining
IT	Information Technology
GP	Grade points
GPA	Grade points average
MDP	Markov Decision Processes
TDU	Tokyo Denki University

Chapter 20
Intelligent Decision-Making Support within the E-Learning Process

Dorota Dżega[1] and Wiesław Pietruszkiewicz[2]

[1] West Pomeranian Business School in Szczecin
ul. Żołnierska 53, 71-210 Szczecin, Poland
ddzega@zpsb.szczecin.pl
[2] West Pomeranian University of Technology in Szczecin
ul. Żołnierska 49, 71-210 Szczecin, Poland
wpietruszkiewicz@wi.zut.edu.pl

Abstract. With the advance of the technological capabilities of the Web and the change of learning habits, e-learning has become a real educational choice accounted by many organizations and students. However, this virtual process is harder to manage and use efficiently than traditional learning. This is because of the complexity of e-learning software and the amount of available data, which has outgrown human expectations and processing capabilities. Fortunately, there is a solution: many dimensions of e-learning can be supported by artificial intelligence (AI). In this article we investigate the use of AI in e-learning and analyze particular AI applications, considering their educational as well as business effects. The domain of e-learning is very broad, but its elements might be ordered by their practical importance and the ratios of profits to costs of AI-oriented investments. Hence, we focus on the issues most relevant in practice. This chapter was based on managerial observations of available e-learning courses and the analysis of requirements for applied AI.

20.1 Introduction

The popularity of e-learning definitively answers the fundamental question of whether we should be using it. Therefore, a new question dominates: How can we make it better? As the number of people using e-learning increases, so too does their expectation of what it can achieve and the complexity of e-learning environments. These environments are harder to observe and control, making it difficult to offer services that meet the users' demands. At this mature stage of development, it seems that the universal solution to these problems is to deploy AI in various e-learning activities. AI could perform time-consuming tasks, offer instant support, or analyze large amounts of data far beyond human capabilities.

We have identified advantages and disadvantages of AI applied to e-learning. These features relate to three areas of the e-learning process – technological,

A. Peña-Ayala (Ed.): Intelligent and Adaptive ELS, SIST 17, pp. 497–521.
springerlink.com

business, and educational – and are set in Table 20.1. It must be emphasized that these features do not relate to e-learning itself, but rather to AI applied in this process.

The disadvantages presented in Table 20.1 require constant review, but importantly, by using proper risk management they can be significantly reduced. Hence, in our opinion, the practical use of AI in e-learning must involve a deep analysis and not focus only on the practical applications of the technology.

The domain of e-learning and applied AI is very broad; however, its elements can be ordered by their practical importance and their ratios of profit to cost of investment. Therefore, we will focus on the issues most relevant in practice. This chapter was made on the basis of observations of currently offered e-learning resources and cost analysis of possibly applied AI.

This chapter starts by investigating the main areas of e-learning in which AI could be effectively deployed. The next sections will provide details of potential issues and about our experiments with AI-oriented software development.

Lastly, we state: e-learning could be performed using various electronic means; however, the most popular and effective kind of e-learning utilizes the Web. Therefore, we focus our analysis of the e-learning process supported by the Web.

Table 20.1 The main advantages and disadvantages of AI-supported e-learning in three areas

Area	Advantages	Disadvantages
Technological	Autonomous mechanisms not requiring constant human observation Supporting mechanisms reduce manual user support as information required is offered by supporting mechanisms	IT environment is more complex and more exposed to potential technical problems Higher deployment costs Higher maintenance costs
Business	Better quality of educational product due to faster identification of potential user problems Product tailored to one's demands via self-adaptive courses Automated analysis of clients' performance and behavior, resulting in more precise manual support	Higher requirements for skilled staffers who have to deal with more advanced systems Heterogeneous environment to control and manage
Educational	Progress of learning adjusted to one's performance General framework adaptable to different courses Learner is more engaged in the process adaptive courses support the idea of memorization as they reintroduce material forgotten or not properly understood	Possible danger of more distant student-teacher relationship, as each side might focus only on technological elements More complex environment to use Not all system-offered suggestions may be acceptable to students Teachers may rely too much on autonomous mechanisms and not fully engage in the process

20.2 The Key Areas in Which AI Is Applied to E-Learning

Researchers who investigate e-learning usually focus on a single aspect; they limit their research to the observation of user behavior, to exposition of didactic material, or on procedures ensuring proper quality levels in distance learning systems (Kay and Lum 2004, Chen 2008, Mandinach 2005, Park et al. 2005, Kacalak et al. 2010, Ehlers et al. 2005). However, it is necessary to remember that organizational features play a very important role in distance learning along with technological and educational components. Those managing modern educational institutions should take into account all the aspects mentioned above. Hence, it is suggested that the best way to analyze e-learning is to include components of the educational subprocess and the business subprocess. We have presented such an analysis in Fig. 20.1 – for simplicity it was assumed that the course content was created in one step and optional further, course-related iterations correspond to proper stages. This cycle equals to a typical project life cycle, which constitutes of a preliminary phase, realization, completion, and final evaluation.

The adoption of the project or product life cycle concept for this domain allows us to initiate this kind of process arrangement. Every course consists of the preliminary, intermediate, and final phases. On the other hand, every course is a part of a given educational status (e.g., curriculum or studies program). The e-learning course life cycle model is complemented by the e-content life cycle model, as it was presented in (Pietruszkiewicz and Dżega 2012).

Fig. 20.1 differs from other papers, as they perceive only educational aspects of the e-learning process. In our opinion, e-learning services, regardless of whether they are offered in universities or for corporate organizations, are a specialized virtual service and their analysis should also incorporate business subprocesses. This is the novelty we investigate in this chapter.

It needs to be emphasized that the success of every educational offer depends not only on its content, but also on a wide range of factors both external (e.g. demographic situation, demands of labor market, trends, and tendencies, etc.) and internal (e.g., manpower management, information infrastructure, organizational, and administrative structures, etc.). Internal factors are controllable, while external ones are often not.

Learning management systems (LMS) generate vast amount of data related to e-learning users. The record and assessment of the students' actions and results, and the analysis of these large datasets is beyond human capabilities. In consequence, AI sometimes uses data mining (DM) techniques in e-learning to find out patterns of behavior and predict outcomes. The survey stated by Baker and Yacef 2009 analyzes the application of AI and DM methods in e-learning based on the *Proceedings of Educational Data Mining* in 2008 and 2009. The results of this work reveal the following statistics:

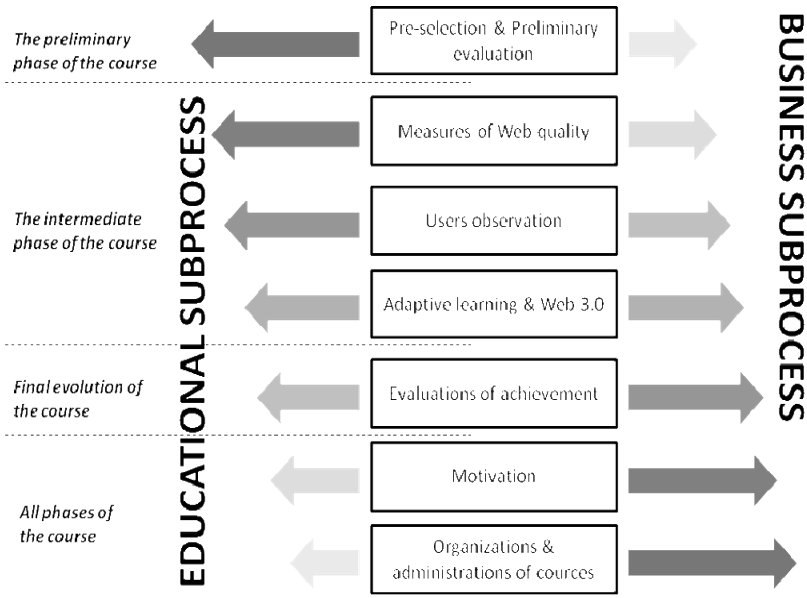


Fig. 20.1 The main components of the e-learning process and its phases

- Relationship mining: 9%,
- Clustering: 15%,
- Prediction: 42%,
- Human judgment/exploratory data analysis: 12%,
- Discovery with model: 19%,
- None of the above: 28%.

It must be noted that the numbers do not add up to 100%, as some papers used multiple methods that were counted in multiple categories.

The largest group of AI applications within e-learning is aimed at observation, control, and predication of user behavior or performance. There are various papers presenting these aspects of AI in e-learning. For example, (Yu et al. 2001) analyzed how decision trees may be used by a teacher to find relationships between students' marks and their activity. The work of (Delgado Calvo-Flores et al. 2008) deploys neural networks to predict students' marks, and (Romero and Ventura 2007) describes a comparison of different classifiers used as student grades predictors. Others researches relate the analysis of AI procedures to their use in e-learning courses with the aim to identify their improvement (Baruque et al. 2007), and make a qualitative evaluation of e-learning courses (Balogh 2009). AI is also successfully used in a case-based reasoning application for distance learning (Shen et al. 2003).

Among other AI methods used in e-learning, it is worth mentioning that clustering was used to analyze the patterns of courses used by students (Ventura 2008) and to analyze the virtual usage steps (Mor and Minguillón 2004).

Another approach is the creation of an automated mechanism (Markellou et al. 2005) that recommends relevant content in students and then analyzes the students' learning sequences (Pahl and Donnellan 2002). A detailed survey of educational data mining was done by (Romero et al. 2008).

The next group of AI applications within e-learning contains those related to adaptive e-learning courses. The majority of popular courseware systems use a static approach to e-learning course design. This approach has three main shortcomings (Brusilovsky and Nijhavan 2002):

1. Modern reusability frameworks implicitly assume that a learning object is a moveable entity,
2. The resource repositories are being constantly updated,
3. The "one-size-fits-all" problem cannot be solved by static courses.

As a potential solution to these problems, the Knowledge Tree was proposed by (Brusilovsky 2004). Its main goal is to bridge the gap between the currently popular approach, which is centered on learning management systems, and Web-based education using intelligent tutoring and adaptive hypermedia. The Knowledge Tree is a distributed architecture for adaptive e-learning and was based on the reuse of intelligent educational activities, such as: activity servers, value-adding services, learning portals, and student model servers.

In another paper, (Shute and Towle 2003) proposed adaptive learning as a model of four components: content model, learner model, instructional model, and adaptive engine. The content model uses two types of requirements: one group for the delivery system and another for the learning content that will be delivered. The learner model contains information originating from assessments. The instructional model assumes active construction of knowledge as well as realistic and complex problem-solving tasks. The fourth component identifies the student's knowledge and matches learning objects based on this information.

20.3 Phase–Dependent E-Learning Items Supported by AI

The e-learning process contains three major phases i.e., preliminary, intermediate, and final evaluation. All of them contain different components that could be supported by AI – preselection and preliminary evaluation, measures of Web quality, user observation, adaptive learning, Web 3.0, and evaluations of achievements.

20.3.1 Preselection and Preliminary Evaluation

Preliminary tests conducted among learners before the beginning of a course are the most popular means of estimating initial knowledge and skills. This tool is most commonly used in language education.

It is worth mentioning that pretests not only make possible to evaluate a learner prior knowledge, but also help one understand his or her track of reasoning and observe the first signs of user behavior. At this initial stage of preliminary evaluation, if conducted online it is possible to spot incorrect patterns of answers and to use these to generate a personal learning path (Chen 2008). These personal learning paths influence both educational and business fields within the institution that conducts the e-learning course. Table 20.2 presents AI-supported activities fitting into preselection and preliminary evaluation phases.

20.3.2 Measures of Web Quality

E-learning in its most mature variant – for instance, at a higher academic level – relies strongly not only on courses available via LMS, but also on various materials available on the Web. These are provided by teachers, fellow learners, or found by learners themselves. However, full reliance on information found in the Web requires an analysis of their quality (this applies to printed materials as well), as there are major disadvantages to Web material:

- Quality of content: It is easy to publish Web material without any checks and with no accountability,
- Accuracy: Information presented on web pages might not be true; this applies also to sites connected to valid web pages,
- Relevance: Even high-quality material is useless if is not relevant to the subject, and there is little control over content tagging or given titles.

These problems create significant concerns when the Web is treated as a reliable and trusted source of information. A more detailed analysis is found in (Calero et al. 2001, Kitchenham and Charters 2007, Markellou et al. 2005, or Moraga et al. 2009). Table 20.3 presents the implications of Web quality estimation in both e-learning subprocesses.

Table 20.2 Subprocesses in preselection and preliminary evaluation and matching AI-supported activities

Subprocesses	AI-supported activities
Educational subprocess	Identification of reasons for entering the course and the initial level of motivation; adaptation of the curriculum to the learners' initial knowledge levels
Business subprocess	Planning of an educational offer suitable to the learners' needs; improvement resources planning, including teachers

Table 20.3 Subprocesses in measures of Web quality and matching AI-supported activities

Subprocesses	AI-supported activities
Educational subprocess	Higher-quality material used by the learner results in a better understanding of a course subject. Users are advised to use different learning materials, including those they found themselves
Business subprocess	Verification of materials linked to the courses and support for the LMS administrative staff

Quality estimation might be and usually is done manually. But it requires additional resources, is time consuming, and due to compliance is rejected or forgotten by users unaware of its importance. To overcome these problems, we have proposed an automated mechanism supported by AI in form of a black box. This model could be used by an unskilled client. Moreover, this idea fits into a general vision of Web 3.0.

This is understood to be the next stage in Web advancement, orientated to fulfilling users' informational requirements by software intermediating between user and informational sources. In this vision of the Web, the user while browsing is supported by various software agents performing helping tasks such as searching or filtering. The general schema for this proposed method is presented in Fig. 20.2. This software was part of a SDART Ltd. project completed in cooperation with the West Pomeranian Business School.

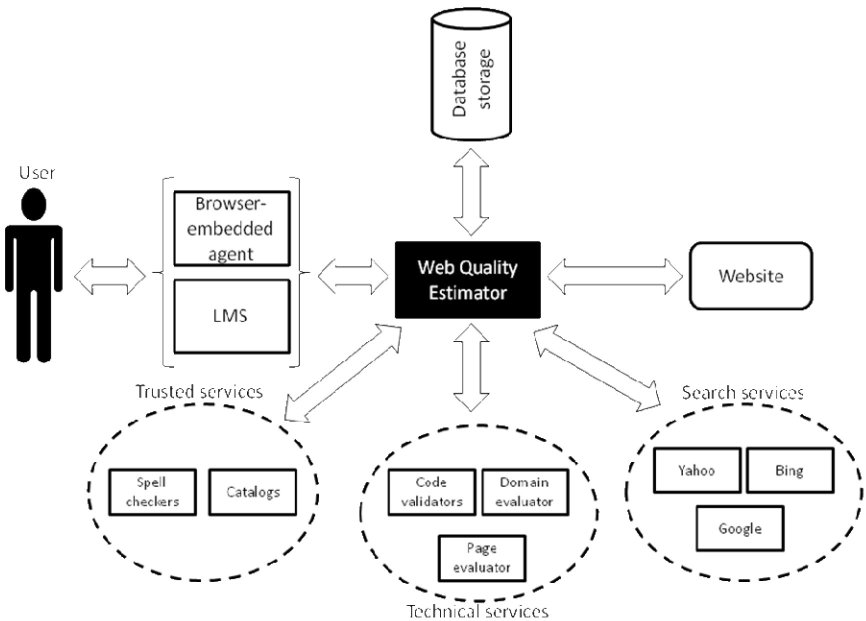


Fig. 20.2 The general schema of Web Quality Estimator

There are two ways to use the Web Quality Estimator (WQE), both of which require software wrappers allowing it to deploy its functionality outside the research and development computer environment:

- As an additional LMS module: being developed as a part of Moodle LMS. In this case the LMS evaluates posted or attached URLs and the LMS user is then informed about their potential value,
- As a browser-embedded agent: being developed as a XUL-based plug-in for Mozilla browsers. This solution is more convenient, as it works not only in the

LMS environment, but also in any other situation. Each visited page might be evaluated and while browsing, the user receives immediate feedback.

These methods were selected to provide an environment for the user in situations in which intelligent supporting mechanisms will indeed offer support by observing the user in his/her natural Web environment. Additional standalone observation applications were rejected; although they could be successfully used in a research environment, they are burdensome in a production-ready environment.

Table 20.4 presents groups of WQE features used in the evaluation including technical, environmental, and external (all also present in Fig. 20.3). These groups represent three major areas of website quality relating to its content, technical preparation, and connections with other materials.

There are two stages of WQE creation, as it is illustrated in Fig. 20.3:

- Training: This is a machine-learning task. The classifier must be trained using a set of features extracted from the top-100 webpages returned by three major search engines, Google, Yahoo, and Bing. There are 100 different search criteria in which class attributes represent the position of the page in ranking, varying from 0 (top-10 pages) to 9,
- Simulation: The trained WQE embedded inside a software mechanism is ready to be used and evaluate new pages (as mentioned above, the plan is to deploy it in a LMS module and browser agent).

Table 20.4 Website quality features

Group	Features	Meaning
Technical	Relating to the page	Well-functioning page with a small number of valid outgoing links and well structured
	Relating to the domain	Trusted top-level domain or a popular commercial domain
	Relating to the webpage	Technically correct page (according to Web standards) with additional information in meta-tags.
Environmental	Based on search engines' evaluation	Top ranked by the search engines, with a large amount of ingoing links (on highly ranked pages)
External	Based on information evaluation done via trusted services	Page or domain present in well-known catalogues and with error-free spelling.

Fig. 20.3 The stages of WQE usage

There are two stages of WQE creation, as it is illustrated in Fig. 20.3:

1. Training: This is a machine-learning task. The classifier is trained using features extracted from the top-100 webpages returned by three search engines, Google, Yahoo and Bing. There are 100 criteria in which class attributes represent the position of the page in ranking, varying from 0 to 9 (top-10 pages),
2. Simulation: The trained WQE embedded inside a software mechanism is ready to be used and evaluate new pages (as mentioned above, the plan is to deploy it in a LMS module and browser agent).

It must be explained why search engines are used as a source of rank attributes. Modern search engines use weighting mechanisms to measure page importance, relating it to other pages, and using links joining all the networks of hyperlinked documents. Hence, the position of a webpage is influenced by its popularity among other pages, as well as by their strength. In short, the more and better pages links for a webpage, the position of the webpage is higher. More information about hyperlink-based rankings can be found in (Markov and Larose 2007). Search engines are not dull tag analyzers (although they are not resistant to tag manipulation), as they were in their first generation. They use website popularity and are influenced by the quality of the website, which itself is determined by an evaluation done by users and links created by them. It is impossible to reveal an algorithm deciding if a particular user will link to a page or not, but it is possible to recreate it using machine learning involving the outcome of overall linking actions.

Such an intelligent mechanism supports the user in browsing and, due to its observing character, does not require any input from the user, easing the difficulties of Web quality analysis. The main risk of this mechanism deployment is overreliance by the user, who might reject or omit interesting and relevant web pages due to their estimated low quality.

20.3.3 Users Observation

User-behavior observation is one of the crucial elements supporting the proper operation of e-learning. It must be remembered that not all users work systematically. In fact, the vast majority of LMS users (both students and teachers) intensify their activity in particular circumstances – for instance, shortly before an exam, or at the beginning or end of a course.

For this reason it is extremely difficult to assess the efficiency of online work performed by each user based only on the final or periodic grades (grade that learners receive, e.g., at the end of each section). The users' awareness that they are supervised in most cases is encouraging and makes them work more systematically. Hence, it is necessary to provide proper tools that would enable the observation of users in the areas defined in Table 20.5.

It is by no means unusual in traditional learning that didactic material differs depending on learner ability. However, in distance learning, standardization of didactic material has taken place. While valuating this, one must consider the fact that attention was paid to make multimedia material more attractive rather than to meeting the needs of particular users, while also recognizing that investments in e-learning technologies remain very expensive and many institutions that offer distance learning cannot afford to prepare different versions of the same course. In (Kay and Lum 2004), authors highlighted the necessity of providing solutions, including profiled actions designed for the users according to their needs. For instance, it is desirable to offer additional hints and guidelines for users whose progress in learning is slower. Those users who are quicker should have an opportunity to gain access to additional knowledge support (i.e., to extended materials, presentations, or exercises).

User behavior observation is of major importance when it comes to spotting events or facts that signal a high likelihood of disturbance in the learning process, which may even lead to the discontinuance of course attendance. On the basis of our experience, the following indicators of course neglect may be identified: sense of isolation; a teacher's instructions are not readily comprehensible; didactic material that is not adapted to the learner's needs; difficulties in dealing with the course's interface; and feedback that is not detailed enough.

While observing user behavior, it is crucial to make every effort to spot the first signs that may indicate the forthcoming discontinuance of the course. However, the notion of discontinuance of the course discussed herein refers to "dropout students (or non-completers) as students that voluntarily withdraw from e-learning courses while acquiring financial penalties" – as defined in (Levy 2007), and it should not be identified with the situation in which the participant failed to complete the course successfully or was asked to no longer attend.

Although discontinuance of e-learning courses has been of interest to researchers – e.g., (Hämäläinen and Vinni 2006, Hammouda and Kamel 2007, Kotsiantis et al. 2003, Lykourentzou et al. 2009), there is still no solution in sight. To conduct a complete observation of users, more effort is required than just the reporting of *actions* taken by users in LMS.

Table 20.5 Subprocess in user observation and matching AI-supported activities

Subprocess	AI-supported activities
Educational	Spotting the signals of the fall of the students' activity; maintenance of the proper level of activity; suggestions on planning and realization of the learning process
Business	Spotting the signals of a fall in a teacher's activity; development of manpower

The main duties and responsibilities of students are: their participation in the course and the response to tests that evaluate their knowledge and skills. Whereas, the main duties and responsibilities of teachers are: planning and conducting the course, checking the students' knowledge, assessment, and filing of documents. Hence, it is necessary to include the dimension: *dutifulness*. Both dimensions actions and dutifulness are presented in Fig. 20.4. Knowing the levels of motivation and dutifulness enables educational institutions to adopt the right strategy:

- Elimination of inactive users and persons who do not fulfill their tasks,
- Strengthening users in activity areas; this applies to those users whose activity was low or medium, whereas dutifulness was medium or high,
- Strengthening users in dutifulness areas; this applies to those users whose activity was medium or high, whereas dutifulness was low or medium,
- Maintenance of engagement; this applies to users whose activity and dutifulness are both high.

The strategies of strengthening are intensified actions in the area of special support, designed for students and teachers. All these strategies are tightly coupled with the component *motivation*, which is presented in the late part of this article.

There exist many successful AI applications to control learners, but there is no method to control the educators/tutors/teachers. As quality maintenance requires observation and control of all users, we have created an AI-supported method that could provide information about user behavior, including that of teachers.

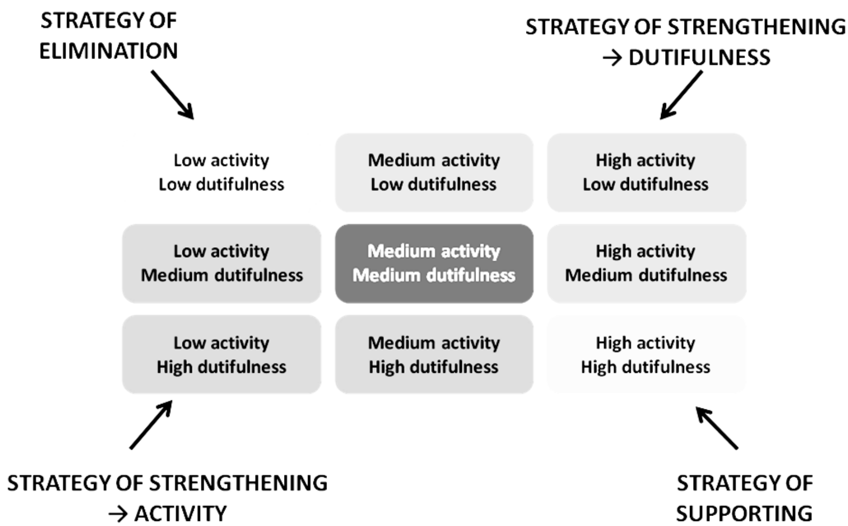


Fig. 20.4 Activity and dutifulness: two dimensions of user behavior

It must be explained why and how difficult it is to manage teachers. For comparison, students have exams, solve exercises, and finally receive grades. Unfortunately, such information is unavailable for teachers. A common approach to evaluate teachers is via questionnaires filled out by students. But questionnaires are expensive, periodic, and it might be uncomfortable for students to provide negative opinions. We wanted to avoid these limitations, and the evaluation method had to only use the data available from LMS.

Fig. 20.5 contains a general schema of this method. As can be seen, the source of information is the LMS logs. They are used to extract necessary attributes that could be used to evaluate user activity. To fully understand this approach, the notion of activity must be explained. An indisputable definition of activity in the e-learning process does not yet exist; therefore, we have created two joined criterions that define a perception of the activity notion (Pietruszkiewicz and Dżega 2012):

1. The variety and number of actions logged for a teacher positively stimulate the value of teacher's activity,
2. Equally spread events are more desired than events grouped in short-time groups.

Due to a missing activity (usage performance) attribute, the users may be grouped into several clusters and later joined into classes. Finally, using created clusters, users can be identified as having one of several activity profiles, for instance, active, moderate, and passive (Fig. 20.6 shows the outcome of this analysis). Use of this method allows managerial staff to frequently analyze the available data and identify potential problems, or to identify users not sufficiently devoted to e-learning.

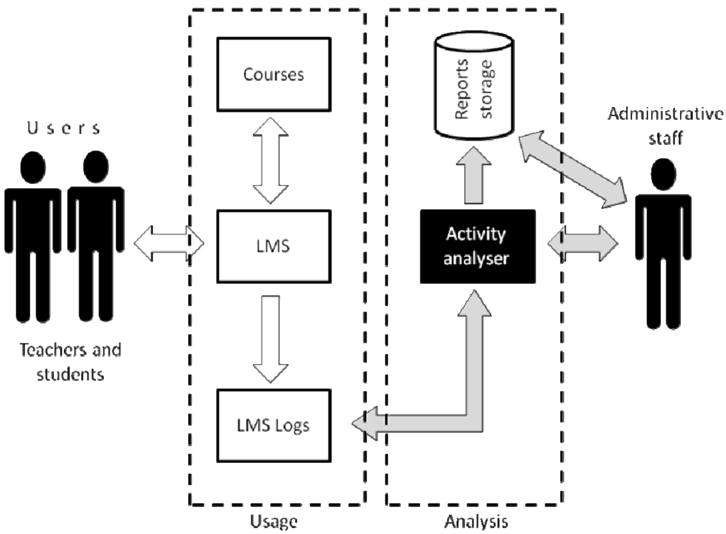


Fig. 20.5 The schema of a method for user evaluation

Understandably, this feature might not be liked by all teachers; however, it must be integrated into common practice, as an assumption that activity evaluation shouldn't be done is as naive as excluding evaluation in a traditional learning environment. Moreover, this method, when compared to traditional learning observations, allows more discreet and effective evaluation. Additionally, the frequency of evaluations helps to reduce a potentially large problem developing unnoticed. Finally, this feature does not differ from work time or access evidence systems, hence teachers, as do other employees, have to embrace it.

The reports containing logged events were divided into six groups representing the six areas of user activity: “Add,” “All actions,” “All changes,” “Delete,” “Update,” and “View.” It was necessary to separate activities into different dimensions, as the proposed solution should provide information not just about overall LMS usage, but also about potential issues (e.g., users may not use LMS actively due to a lack of skills).

To understand the process of activity clustering, we analyze Fig. 20.6. The reports generated by Moodle are typical Web logs and contain logged information about the time of events, but no information about the users’ activity taking place between the two events. An unsuitable approach to this problem might be the calculation of the average numbers of events. However, this would not distinguish frequently working users from those generating large numbers of events in short-time sessions.

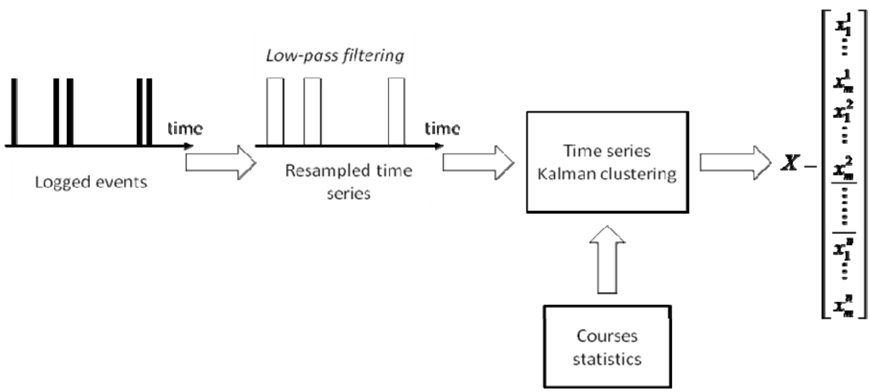


Fig. 20.6 The process of clustering for user observation

This explains why low-pass filtering was required, and this step resampled data in a discrete time window (see Fig. 20.7 for an example; vertical bars represent the number of the course for the user in this time, and the data series represents one of activity components).

Resampled time series data were passed through dynamic system and clustering was completed by a developed algorithm based on a Kalman filter. Together with the course’s statistics, they were used to estimate (via Kalman filtering) clusters parameters (denoted as *X* state variables vector).

By introducing different activity components, it was possible to distinguish LMS “observers” from “writers” – as noted in Fig. 20.8, presenting real-life evaluation of e-learning teachers completed for over 100 courses offered at the West Pomeranian Business School in Szczecin – ZPSB (Polish official abbrev.). More about this approach, performed experiments, and practical implications can be found in Pietruszkiewicz and Dżega 2012.

In summary, this system allows managerial staff to easily observe user behavior in a complex virtual environment. Additionally, to ensure high quality this approach used on a daily basis should be periodically supported by traditional evaluation, such as questionnaires and conversations with users.

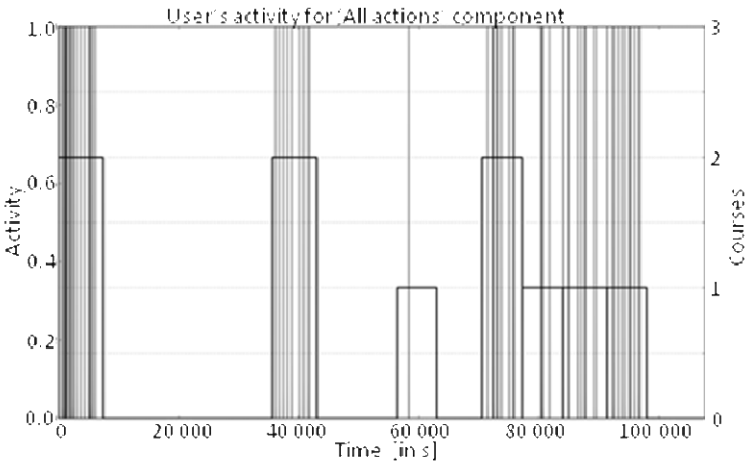


Fig. 20.7 A sample of activity time series

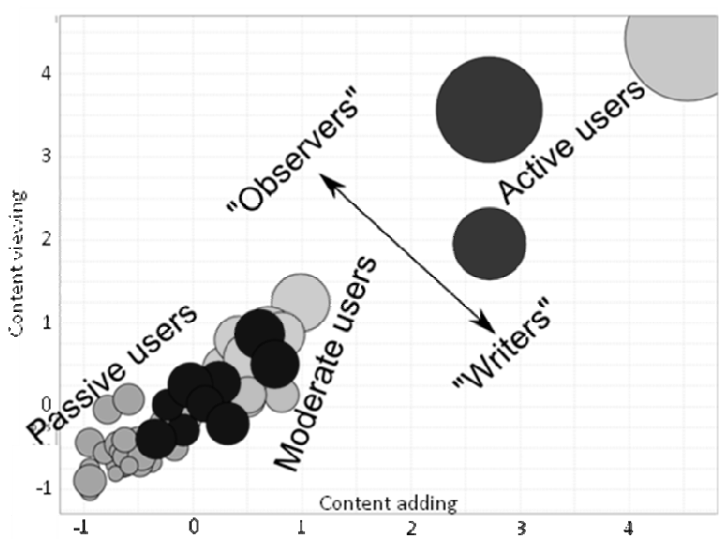


Fig. 20.8 The results of activity clustering completed on real-life samples of teachers' behavior

20.3.4 Adaptive Learning and Web 3.0

According to (Shute and Towle 2003) three different levels of knowledge exist:

1. Basic knowledge: an elementary knowledge about definitions, formulas, and rules. This could be provided by screens,
2. Procedural knowledge: knowledge of how to do something. This could be delivered by interactive slides and examples explaining how different knowledge elements work together,
3. Conceptual knowledge: this is top-level knowledge of relations between several pieces of knowledge and having a view on a subject. It might be explained by summaries and linking screens to create a hierarchical course structure.

E-learning courses should provide pieces of knowledge at each escalating level of understanding. To increase the ratio of remembered and understood material, there must be a functionality to verify and if necessary reintroduce some parts of a course. This has yet to be introduced routinely. The educational and business effects of adaptive course deployment are shown in Table 20.5.

Table 20.5 Subprocesses in adaptive learning and Web 3.0 and matching AI-supported activities

Subprocessesies	AI-supported activity
Educational subprocess	Mechanisms of the maintenance of learners' attention
Business subprocess	Generation of learning statistics for better control of users

It might be confusing as to why this approach (including adaptive courses) is necessary when intelligent tutoring systems (ITS) already exist such as CTAT (Cognitive Tutor Authoring Tools) or Andes. However, some issues and critical opinions about ITS raised by the practitioners are given next:

- There are cost-generating add-ons and their profits returned are questioned,
- They are tailored to a particular course, and this is a resource-consuming task,
- They are hard to deploy in-house, as they significantly increase required levels of skills for the e-learning development team.

Additionally, in its most simplified form ITS may be partially substituted by detailed comments to answers, which is a standard feature offered by LMS.

Hence, according to the Pareto principle commonly used in practice, we were looking for a simpler solution (thus cheap and easy to use) while still solving the majority of challenges. Our main aim is to control the progress of learning and to adjust the presented materials (not evaluate answers). The adjustment of materials should be done by a cost-effective analysis of answers and using inter-linked course materials – a feature introduced at the course development stage. In the results, after the linkage such adaptation mechanisms should be autonomous.

The framework for this mechanism is the SDART Presentation Engine (SPE). This is a course presentation engine developed by SDART Ltd. The SPE was selected as the basis of extensions for several reasons:

- It uses easily modifiable XML files representing courses materials,
- It was build using Flex technology and supports the idea of prototyping,
- SPE uses a modular and easily extensible architecture.

Currently, SPE functionality has been extended (see Fig. 20.9) by designing and developing two additional modules. The first is a learning analyzer, which observes a user's learning progress. It stores information about sections viewed and checks retention by analyzing related questions, and also analyzing the time of actions done and their influence on user memory and knowledge. The second developed module is a course manager, which guides users through the course by proposing slides to learn and verify the previously presented knowledge.

The main idea of adaptive e-learning courses (see Fig. 20.10) is to create an intelligent, self-adjusting mechanism that, apart from presenting the content, will also store the results of learning and analyze them to suggest relevant information and guide users through the courses over a non-linear path. For comparison, in traditional e-learning the flow of courses is linear, as slides form an ordered list and a user moves from one slide to the next neighboring one. On further analysis of Fig. 20.10, we notice a hierarchical logical structure containing four layers:

- Course: an overview layer,
- Chapters: present coherent pieces of information, and they are the major distinguishable elements of each course,
- Sections: form chapters, and they are minor elements relating to precise information, and may use single or a couple screens,
- Tests: they belong to a chapter, via linking to the sections to allow the control of assessment of learning for these elements.

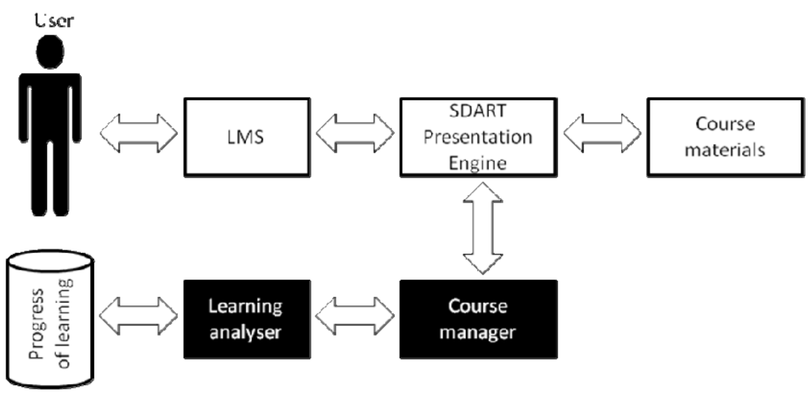


Fig. 20.9 The general schema of an adaptive course mechanism for the SDART Presentation Engine

The distinguishable features of our proposed adaptive courses are the relationships between its elements (i.e., questions linked to sections or the single screens they verify). Using this linkage, it is possible to verify the remembered material by correctly answering questions relating to those particular sections.

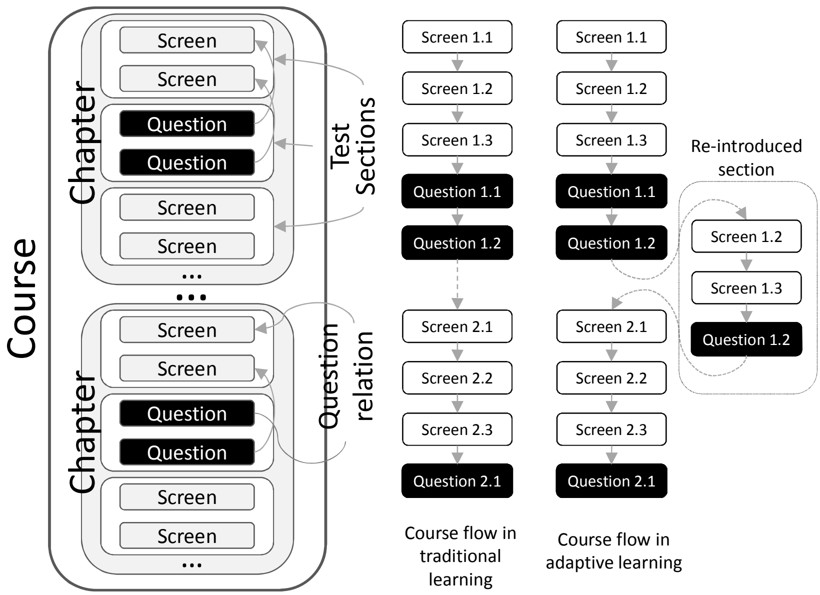


Fig. 20.10 The adaptive courses and their structure

As noted at the beginning of this section, adaptive courses increase the quality of learning by ensuring correct memorization of learned material by reintroducing sections forgotten or incorrectly understood. However, similarly to what we did for the previous sections, we must note that in certain situations this solution might be risky or burdensome.

This is especially valid for users not used to working with supporting programs or users with a strong self-opinion (even if it is incorrect) – i.e., in this case knowing better than the system what they should or should not learn.

20.3.5 Evaluations of Achievements

The evaluation of a learner's achievement is a complicated process regardless of how it is conducted, classroom or online. This task requires the next definitions:

- The object of evaluation (i.e., "what will be assessed?"): Among other things it includes theoretical knowledge, the ability to solve problems, creativity in problem solving, or accuracy in fulfilling a task,

- The subject of the evaluation (i.e., "who will be assessed?"): This might be a learner, student, or employee participating in the training.

For the online evaluation, one must bear in mind that e-learning requires different resources and infrastructure than traditional learning. For that reason, a simple transformation of the paper/pencil test taken in the classroom into an e-learning test is not effective – as stated in (Mandinach 2005) and (Park et al. 2005). It is necessary to take into account areas such as institutional infrastructure, educational or teacher processes, and student learning processes (Mandinach 2005). It is also required to consider the different levels of learners' knowledge (Park et al. 2005). However, the development of Internet technologies makes it easier. It also enables the initiation of new methods of achievement evaluation, not only with the usual tools of conducting the e-learning tests, but also with the use of voice (Kacalak et al. 2010). The virtual areas of support presented in Table 20.6 confirm again that there is a strong connection between educational and business subprocesses.

Table 20.6 Subprocesses in evaluation of achievement and matching AI-supported activities

Subprocesses	AI-supported activities
Educational	Selection of knowledge verification methods; measurement of learners' knowledge gained during the course; creation of the base of good educational practice
Business	Assessment of the quality of teachers' work and activity; formation of the college quality policy (students, teachers); support of the process of motivating and rewarding teachers

The evaluation of achievement is a vitally important component, both from educational and business points of view. For learners it is a stage that depends on the successful completion of the course. For teachers it is a stage in which the quality of their activity and the services they perform are assessed. Finally, for the educational institution it has a great responsibility, for the documents it issues certify the knowledge and skills of the learners (Mandinach 2005).

20.4 Overall E-Learning Components Supported by AI

There are components that do not depend on the phase of e-learning process (i.e., they exist in the whole process and must be controlled and management during all phases). These components include motivation, organizations, and administrations of courses. They will be introduced and discussed in this section.

20.4.1 Motivation

Motivation plays a important role in every field of education, as it determines particular behaviors and actions. The appropriate level of motivation enables one to achieve a set of goals. Motivation is crucial for the e-learning participants, since only highly and properly motivated persons are capable of successfully completing e-learning courses or studies.

The most common reasons for choosing e-learning as an educational path are: work life (e.g., the possibility of reconciling one's career with education); personal situation (e.g., state of health or the distance between the educational institution and place of residence); family (e.g., child care); and economic migration (e.g., a temporary stay and working abroad). On the basis of the prior reasons, it may be assumed that e-learning attendants are highly motivated. However, the reality is quite different. A high percentage of people resign from courses. Researchers point out: Approximately 25–40% of students withdraw from e-learning courses, which is 10–20% more than traditional courses – see (Frankola 2001, Lykourentzou et al. 2009). In (Lykourentzou et al. 2009) e-learning students were divided into four categories of behaviors and outcomes:

1. Students who registered but never entered the e-learning course,
2. Students who entered the e-learning course and completed a number of sections but decided to drop out completely,
3. Students who completed some of the course sections but decided to discontinue their studies and repeat the course in a following semester,
4. Students who completed all of the course sections and successfully completed the course.

Such division of students is reflected in practice. To the first category belong students whose level of motivation before enrolling in the course was not sufficient. While registering, they were driven by impulse, fashion, or some kind of short-term and quickly passing sense of duty. For the students who belong to the second and third categories, a medium level of motivation is characteristic, and it is possible, to some extent, to persuade them to complete the course by means of some encouraging mechanisms (e.g. discussing the benefits of the course and showing them that it is worth completing it). Finally, to the fourth category generally belong students with a high level of motivation, and who have defined reasons for such behavior. It is worth mentioning that not all students from the fourth category manage to successfully complete the course; however, they try.

An important factor that influences motivation levels is the voluntary decision to participate in a particular course. Students' freedom in this area is usually limited, since the vast majority of courses are compulsory. It is very different when a learner makes his/her own decision to take part in a course. In our research we used the example of the Innovative Academic Enterprise e-learning courses, which were conducted by the ZPSB from November 2009 to April 2011. There were 196 participants per course, including students and teachers of the ZPSB. Since there was no option to take a break and return to the course in the future, an analysis of this aspect was not conducted.

Table 20.7 presents the percentage of participants in four categories. The high percentage (almost 50%) of those participants who registered voluntarily and never entered the course indicates that educational institutions that offer e-learning courses might face serious problems (e.g., time, effort, costs, etc., that will not pay off). That is why the preselection of candidates that was discussed in the component preselection and preliminary evaluation is of great importance. Looking at motivation from a business point of view enables us to see and consider both learners and teachers as clients of the educational institution (see Table 20.8). Fig. 20.11 presents various levels of motivation in connection with the orientation of the client: external (learners) and internal (teachers). As it can be noticed in Fig. 20.11, there are two situations that should be allowed in the distance learning process: looking at the client and the organization.

Table 20.7 Percentage of participants of the Innovative Academic Enterprise e-learning courses

Category	Course1	Course2	Course3	Course4*
Participants who registered but never entered the course	45.92%	59.69%	49.49%	35.20%
Participants who entered the course and completed a number of sections but decided not to take the final test	23.47%	12.24%	14.29%	11.22%
Participants who completed all of the course sections but failed to pass the final test	1.02%	1.53%	3.06%	3.06%
Participants who completed all of the course sections and passed the final test	29.59%	26.53%	33.16%	50.51%

* For 44 participants, Course4 was compulsory.

Table 20.8 Subprocesses in the motivation component and matching AI-supported activities

Subprocesses	AI-supported activities
Educational	Support for creating new development projects oriented to learners and teachers; creation of career paths
Business	Identification of client profile; identification of reasons for learners' dropping out; development of manpower management toward teachers

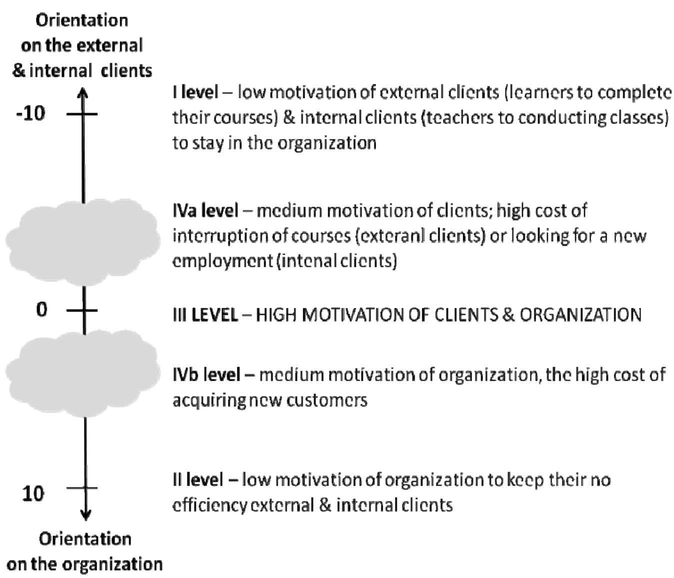


Fig. 20.11 Orientation and motivation in the e-learning process

In both situations we deal with the reality that there is no point in trying to keep a learner or a teacher on at any cost. They are unlikely to appreciate the efforts made by the organization and will probably choose their own way, but in the meantime they may generate high costs.

In the case of learners, these costs would be additional engagement of teachers, the necessity to arrange additional exam dates, an increase in the engagement of administrative workers, communication costs, etc. When it comes to teachers, it has been observed that low attendance during the LMS operating training results in higher costs. This is related to dealing with a particular teacher and his/her increased numbers of errors, failures, and troubles, as ultimately more support is needed. Very often people with a low level of motivation do not change their behavior even after receiving this support.

Similarly, educational institutions are not interested in making the investment required to keep some clients. However, while summing up the experience gathered over our research, it remains crucial to spot this right moment at which action should be taken toward the clients with medium motivation levels or toward the organization with a medium level of motivation to keep a particular client.

When taking action in order to keep clients, apart from the educational aspects, one should take into consideration the following data on clients:

- Demographic features: Sex, marital status, place of residence, education, and professional experience,
- External data: The client's situation in the labor market, the fact that some courses and studies are at this moment more popular, or even fashionable, than others, and social mood,
- The client's behavior outside the course room.

While dealing with such differentiation of data, statistical methods seem to be insufficient. AI methods, on the other hand, may prove to be very effective.

20.4.2 Organization and Administrations of Courses

Making decisions in the area of e-learning management is more complex than in traditional learning. This complexity is a result of both learners and teachers operating in a distributed environment. Moreover, changes in distance learning occur quickly and decisions around e-learning must match this. Many educational institutions claim that as far as decision making is concerned, they apply an individual approach depending on the situation (Nash 2005). This is a time-consuming approach. The research in paper Qureshi et al. 2011, based on a literature analysis, looks at the results of various studies, paying special attention to the following problems with planning, initiation, and development of ICTs in education:

- Resistance of people to change,
- Underestimation, lack of awareness and negative attitudes toward ICTs,
- Lack of systemic approach to implementation and lack of follow-up,
- High rates of system non-completion,
- Lack of user-training,

- Lack of administrative and technical end-user support,
- User dissatisfaction with new systems,
- Mismatches between technologies and the context, culture, and work practices.

All the problems mentioned above are usually tightly coupled with the cost of service of distance learning. For this reason centers, where disturbances arise must be identified and then some measures must be taken to minimize their negative impact both on educational and business subprocesses, as shown in Table 20.9.

Distance learning forces institutions to create special units whose job it is to manage the learning process in a distributed environment. The correct completion of the e-learning courses depends not only on the teacher's engagement, but also on the entire e-learning development team.

Table 20.9 Subprocesses in organization and administration of courses and matching AI-supported activities

Subprocesses	AI-supported activities
Educational	Individualized support (technical, related to training) for teachers and students
Business	Support of the learning process management in a distributed environment; verification (classification) of the failures or breakdowns; support of the reporting to institutions of supervision

According to a report from the United Kingdom Department for Education and Employment the main roles the e-learning development team plays are: project leader, instructional designer, courseware designer/author, programmer, graphics designer, and audio-visual coordinator (Lewis and Whitlock 2003). In practice one person may fulfill several roles; for instance, the role of project leader may be connected with the role of instructional designer. It is also necessary to single out people responsible for the organization of the learning process and the LMS administrator. Often the individual members of the team are dispersed geographically. In this case, when other participants of the learning process (learners and teachers) are also separated, there is a tendency toward the deformation of reality when information is conveyed. This is clearly seen when various events or users' errors are reported as LMS malfunctions.

20.5 Conclusions

AI has many possible areas of application in e-learning. They range from the construction of adaptive learning mechanisms that self-adjust to a students' performance, to the management of supporting mechanisms allowing better control of virtual learning environments, to methods of helping with standard business problems (i.e., customer retention).

However, it is crucial to understand the pros and cons of AI application in e-learning. There are many advantages discussed herein, but a complex AI-supported e-learning environment may face potential problems. These are mainly caused by the phenomenon of a complex solution requiring more attention (i.e., a complex structure is more expensive, less fault-free, and requires higher skill levels to run). The analysis of issues and risks is an important feature missing in most papers that focus on the purely technical or scientific aspects of AI applied to various parts of e-learning. They neglect risk sources and possible obstacles that are crucial to recognize in practice.

The areas of e-learning presented in this article were selected according to their practical importance and the possibility of successful improvements by applying AI, and this was based on the analysis of over 100 courses offered within academic establishments.

To summarize, in our opinion, even if there are some issues relating to the usage of AI in e-learning, we provided arguments of when, why, and how it could best be used. It can be integrated to achieve a better quality of e-learning process and bring advantages to educational as well as business goals.

Acknowledgments. The research presented herein was founded by the research grant N115 413240 from the National Science Centre in Poland.

References

Baker, R.S.J.D., Yacef, K.: The state of educational data mining in 2009: A review and future visions. Journal of Educational Data Mining 1(1), 3–17 (2009)

Balogh, I.: Use of data mining tools in examining and developing the quality of e-learning. In: Szucs, A. (ed.) Proceedings of LOGOS OPE, pp. 129–139 (2009)

Brusilovsky, P.: Knowledge Tree: a distributed architecture for adaptive e-learning. In: Proceedings of WWW ALT 2004, pp. 104–113 (2004)

Brusilovsky, P., Nijhavan, H.: A framework for adaptive E-learning based on distributed reusable learning activities. In: Driscoll, M., Reeves, T. (eds.) Proceedings of ELEARN, pp. 154–161 (2002)

Calero, C., Piattini, M., Genero, M.: Method for obtaining correct metrics. In: Proceedings of ICEIS, pp. 779–784 (2001)

Chen, C.M.: Intelligent web-based learning system with personalized learning path guidance. Computers & Education 51, 787–814 (2008)

Delgado Calvo-Flores, D., Gibaja Galindo, E., Pegalajar Jiménez, M.C., Pérez Piñeiro, O.: Predicting students' marks from Moodle logs using neural network models. In: Méndez-Vilas, A., Solano Martín, A., Mesa González, J.A., Mesa González, J. (eds.) Current Developments in Technology-Assisted Education, pp. 586–590. Formatex, Badajoz (2006)

Ehlers, U.D., Goertz, L., Hildebrandt, B., Pawlowski, J.M.: Quality in e-learning: Use and dissemination of quality approaches in European e-learning: a study by the European Quality Observatory (2005), `http://www.voced.edu.au/content/ngv39172` (accessed April 10, 2011)

Frankola, K.: Why online learners dropout. Workforce 80(10), 53–63 (2001)

Hämäläinen, W., Vinni, M.: Comparison of Machine Learning Methods for Intelligent Tutoring Systems. In: Ikeda, M., Ashley, K.D., Chan, T.-W. (eds.) ITS 2006. LNCS, vol. 4053, pp. 525–534. Springer, Heidelberg (2006)

Hammouda, K., Kamel, M.: Data mining in e-Learning. In: Pierre, S. (ed.) E-learning Networked Environments and Architectures: A Knowledge Processing Perspective, pp. 374–404. Springer, London (2007)

Kacalak, W., Majewski, M., Zurada, J.M.: Intelligent E-Learning Systems for Evaluation of User"s Knowledge and Skills with Efficient Information Processing. In: Rutkowski, L., Scherer, R., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) ICAISC 2010. Lecture Notes in Artificial intelligence, vol. 6114, pp. 508–515. Springer, Heidelberg (2010)

Kay, J., Lum, A.: Building User Models from Observations of Users Accessing Multimedia Learning Objects. In: Nürnberger, A., Detyniecki, M. (eds.) AMR 2003. LNCS, vol. 3094, pp. 36–57. Springer, Heidelberg (2004)

Kitchenham, B., Charters, S.: Guidelines for performing systematic literature reviews in software engineering. Technical Report, Keele University and University of Durham (2007)

Kotsiantis, S.B., Pierrakeas, C.J., Pintelas, P.E.: Preventing Student Dropout in Distance Learning using Machine Learning Techniques. In: Palade, V., Howlett, R.J., Jain, L. (eds.) KES 2003, Part II. LNCS (LNAI), vol. 2774, pp. 267–274. Springer, Heidelberg (2003)

Levy, Y.: Comparing dropouts and persistence in e-learning courses. Computers & Education 48(2), 185–204 (2007)

Lewis, R., Whitlock, Q.A.: How to plan and manage an e-learning programme. Gower Publishing Limited, Aldershot (2003)

Lykourentzou, I., Giannoukos, I., Nikolopoulos, V., Mpardis, G., Loumos, V.: Dropout prediction in e-learning courses through the combination of machine learning techniques. Computers & Education 53, 950–965 (2009)

Mandinach, E.B.: The development of effective evaluation methods for e-learning: A concept paper and action plan. Teachers College Record 107(8), 1814–1835 (2005)

Markellou, P., Mousourouli, I., Spiros, S., Tsakalidis, A.: Using semantic web mining technologies for personalized e-learning experiences. In: Uskov, V. (ed.) Proceedings of WBE, pp. 522–527 (2005)

Markov, Z., Larose, D.T.: Data mining the web: Uncovering patterns in web content, structure and usage. John Wiley & Sons, Inc., New Jersey (2007)

Mor, E., Minguillón, J.: E-learning personalization based on itineraries and long-term navigational behavior. In: Proceedings of WWW ALT 2004, pp. 264–265 (2004)

Moraga, C., Moraga, M.A., Caro, A., Calero, C.: SPDQM: SQuaRE-aligned portal data quality model. In: Rossi, G., Iturrioz, J. (eds.) Proceedings of QSIC, pp. 117–122 (2009)

Nash, S.S.: Leadership and the e-learning organization. Texture Press, New York (2006)

Pahl, C., Donnellan, C.: Data mining technology for the evaluation of web-based teaching and learning systems. In: Proceedings of ELEARN, pp. 747–752 (2002)

Park, S.T., Byun, D.W., Park, D.W., Lee, H.: Evaluation system in e-learning through the knowledge state analysis method (accessed March 24, (2005), `http://faculty.ksu.edu.sa/Shammami/Documents/CSC%20541/Paper/370.pdf`

Pietruszkiewicz, W., Dżega, D.: The artificial intelligence in the support of e-learning management and quality maintenance. In: Anastasiades, P. (ed.) Blended learning Environments for Adults. IGI Global, Hershey (2012)

Qureshi, Q.A., Nawaz, A., Khan, N.: Prediction of the problems, user-satisfaction and prospects of e-learning in HEIs of KPK, Pakistan. International Journal of Science and Technology Education Research 2(2), 13–21 (2011)

Romero, C., Ventura, S.: Educational data mining: A survey from 1995 to 2005. Expert Systems with Applications 33(1), 135–146 (2007)

Romero, C., Ventura, S., Espejo, P.G., Hervás, C.: Data mining algorithms to classify students. In: Proceedings of EDM, pp. 8–17 (2008)

Shen, R., Han, P., Yang, F., Yang, Q., Huang, J.: Data mining and case-based reasoning for distance learning. International Journal of Distance Education Technologies 3(1), 46–58 (2003)

Tang, T.Y., McCalla, G.: Student modeling for a web-based learning environment: A data mining approach. In: Dechter, R., Kearns, M., Sutton, R. (eds.) Proceedings of AAAI/IAAI, pp. 967–968 (2002)

Shute, V., Towle, B.: Adaptive e-learning. Educational Psychologist 38(2), 105–114 (2003)

Ventura, S., Romero, C., Hervás, C.: Analyzing rule evaluation measures with educational datasets: A framework to help the teacher. In: Baker, R., Barnes, T., Beck, J. (eds.) Proceedings of EDM, pp. 177–181 (2008)

Abbreviations

AI	Artificial Intelligence
CATS	Cognitive Tutor Authoring Tools
DM	Data Mining
ICT	Information and Communications Technology
ITS	Intelligent Tutoring Systems
LMS	Learning Management System
SPE	SDART Presentation Engine
URL	Uniform Resource Locator
WQE	Web Quality Estimator
XML	Extensible Markup Language
XUL	XML-based User-interface Language
ZPSB	West Pomeranian Business School

Author Index